# Robust image classification: analysis and applications

THÈSE N$^{O}$ 7258 (2016)

PRÉSENTÉE LE 16 DÉCEMBRE 2016

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

LABORATOIRE DE TRAITEMENT DES SIGNAUX 4

PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Alhussein FAWZI

acceptée sur proposition du jury:

Prof. P. Vandergheynst, président du jury
Prof. P. Frossard, directeur de thèse
Prof. J. Bruna, rapporteur
Prof. N. Paragios, rapporteur
Dr F. Fleuret, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2016

# Acknowledgements

I would like to thank my PhD advisor, Prof. Pascal Frossard, for giving me the opportunity to do a PhD in his research group. I am grateful for his guidance and for giving me the freedom I needed to pursue my research interests. Besides, I was very fortunate to benefit from his experience and insights. I also greatly thank him for providing thorough and detailed feedbacks. I finally thank him for his constant support and consideration, and for always believing in me!

I would also like to thank the members of my thesis committee, Prof. Joan Bruna, Dr. François Fleuret, Prof. Nikos Paragios, and Prof. Pierre Vandergheynst for the time they have taken to review my manuscript, and for their helpful comments.

I would like to thank my colleagues and collaborators from IBM Ireland and Watson; Mathieu Sinn, Bei Chen, Jean-Baptiste Fiot, Horst Samulowitz, Deepak Turaga, as well as all others that I enjoyed talking to during the internships. Many works would not have been possible without their precious help and support. Thanks to Olivier Verscheure for giving me the opportunity to do these internships. I have really enjoyed the many discussions we've had, and his constant motivation has always been a great support. Special thanks to Prof. Mike Davies and Prof. Laurent Jacques for the enriching discussions, and for inviting me to Edinburgh and Louvain-La-Neuve.

I also sincerely thank all the current and former members of the lab Ana, Andreas, Beril, Chenglin, David, Eirini, Elif, Ersi, Francesca, Hermina, Jacob, Laura, Luigi, Nikos, Renata, Pinar, Sofia, Tamara, Thomas, Vijay, Xiaowen. I thank Dorina for sharing the office with me during all those years, Stefano and Mattia for the very exciting and entertaining discussions, and Seyed for being a great colleague and a genuinely nice person I enjoyed collaborating with.

Finally, I would like to thank many friends, especially from GMU, who made life in Lausanne very enjoyable. I also greatly thank my brothers, Omar and Hamza, for their constant help and their continuous support. And it is a pleasure to thank my parents and my wife for everything.

# Abstract

In the past decade, image classification systems have witnessed major advances that led to record performances on challenging datasets. However, little is known about the behavior of these classifiers when the data is subject to perturbations, such as random noise, structured geometric transformations, and other common nuisances (e.g., occlusions and illumination changes). Such perturbation models are likely to affect the data in a widespread set of applications, and it is therefore crucial to have a good understanding of the classifiers' robustness properties. We provide in this thesis new theoretical and empirical studies on the robustness of classifiers to perturbations in the data.

Firstly, we address the problem of robustness of classifiers to *adversarial* perturbations. In this corruption model, data points undergo a *minimal* perturbation that is specifically designed to change the estimated label of the classifier. We provide an efficient and accurate algorithm to estimate the robustness of classifiers to adversarial perturbations, and confirm the high vulnerability of state-of-the-art classifiers to such perturbations. We then analyze theoretically the robustness of classifiers to adversarial perturbations, and show the existence of learning-independent limits on the robustness that reveal a tradeoff between robustness and classification accuracy. This theoretical analysis sheds light on the causes of the adversarial instability of state-of-the-art classifiers, which is crucial for the development of new methods that improve the robustness to such perturbations.

Next, we study the robustness of classifiers in a novel *semi-random* noise regime that generalizes both the random and adversarial perturbation regimes. We establish precise theoretical bounds on the robustness of classifiers in this general regime, which depend on the *curvature* of the classifier's decision boundary. Our bounds show in particular that we have a *blessing* of dimensionality phenomenon: in high-dimensional classification tasks, robustness to random noise can be achieved, even if the classifier is extremely unstable to adversarial perturbations. We show however that, for semi-random noise that is mostly random and only mildly adversarial, state-of-the-art classifiers remain vulnerable to such noise. We further perform experiments and show that the derived bounds provide very accurate robustness estimates when applied to various state-of-the-art deep neural networks and different datasets.

Finally, we study the invariance of classifiers to geometric deformations and structured nuisances, such as occlusions. We propose principled and systematic methods for *quantifying* the robustness of arbitrary image classifiers to such deformations, and provide new numerical methods for the estimation of such quantities. We conduct an in-depth experimental evaluation and show that the proposed methods allow us to quantify the gain in invariance that results from increasing the depth of a convolutional neural network, or from the addition of transformed samples to the training set. Moreover, we demonstrate that the proposed methods identify "weak spots" of classifiers by sampling from the set of nuisances

that cause misclassification. Our results thus provide insights into the important features used by the classifier to distinguish between classes.
Overall, we provide in this thesis novel quantitative results that precisely describe the behavior of classifiers under perturbations of the data. We believe our results will be used to objectively assess the reliability of classifiers in real-world noisy environments and eventually construct more reliable systems.

Key words: classification, robustness, adversarial perturbations, random noise, semi-random noise, invariance, geometric transformations, nuisance, occlusions, deep learning, convolutional neural networks.

# Résumé

Les systèmes de classification d'images ont récemment connu des avancées majeures qui ont conduit à des performances impressionantes sur des données d'images complexes. Malgré ces avancées, le comportement de ces systèmes lorsque les données subissent des *perturbations*, telles que du bruit aléatoire ou des transformations géométriques complexes demeure mal compris. Ces modèles de perturbations peuvent affecter les données dans de nombreuses applications, et il est donc essentiel d'avoir une bonne compréhension des propriétés de robustesse des classifieurs. Le but de cette thèse est de fournir une analyse théorique et empirique approfondie de la robustesse des classifeurs aux perturbations qui peuvent affecter les données.

Nous abordons dans un premier temps le problème de la robustesse des classifeurs à des perturbations adverses. Les données subissent, dans ce modèle, des perturbations adverses *minimales* nécessaires afin de changer la classe estimée par le classifieur. La première contribution de cette thèse est un algorithme efficace permettant d'estimer la robustesse des classifieurs aux perturbations adverses. Cet algorithme nous permet, entre autres, de confirmer la vulnérabilité des classifieurs modernes à de telles perturbations. Nous analysons ensuite théoriquement la robustesse des classifieurs aux perturbations adverses, et nous montrons l'existence de limites fondamentales sur la robustesse qui révèlent un compromis entre la robustesse et la performance. Cette analyse théorique nous permet de mieux comprendre les causes de l'instabilité de classifieurs vis-a-vis de perturbations adverses, ce qui représente une étape cruciale pour le développement de nouvelles méthodes améliorant la robustesse des classifieurs à ces perturbations.

Nous étudions dans un second temps la robustesse des classifieurs à un nouveau régime de perturbations *semi-aléatoire*, qui permet d'unifier les régimes aléatoires et adverses. Nous établissons des bornes théoriques précises sur la robustesse des classifieurs dans ce régime général, qui dépendent de la *courbure* de la frontière de décision du classifieur. Nos résultats montrent en particulier que nous avons un phénomène de bénédiction de dimensionnalité, car il est possible d'atteindre une grande robustesse au bruit aléatoire en haute dimension, même si le classifieur est extrêmement instable aux perturbations adverses. Nous montrons, cependant, que si le bruit est principalement aléatoire et seulement légèrement adverse, les classifieurs modernes restent vulnérables à un tel bruit. Nous effectuons en outre des expériences montrant que les résultats théoriques établis fournissent des estimations très précises lorsqu'elles sont appliquées à divers réseaux de neurones profonds et à des ensembles de données complexes.

Enfin, nous étudions l'invariance de classifieurs par rapport à des déformations géométriques et nuisances structurées, telles que les occlusions. Nous proposons des méthodes systématiques permettant de *quantifier* la robustesse des classifieurs d'images à de telles déformations, et proposons des algorithmes numériques efficaces permettant d'estimer ces quantités. Nous effectuons une évaluation expérimentale et montrons que les méthodes

proposées permettent de quantifier le gain en invariance qui résulte de l'augmentation de la profondeur des réseaux de neurones, ou de l'addition d'échantillons transformés aux jeu de données d'apprentissage. De plus, nous démontrons que les méthodes proposées permettent de découvrir les "failles" des classifieurs à l'aide d'un échantillonage dans l'ensemble des nuisances. Enfin, nos résultats fournissent des indications sur les caractéristiques importantes utilisées par un classifieur afin de distinguer entre les classes.
En résumé, nous fournissons dans cette thèse de nouveaux résultats quantitatifs qui décrivent précisément le comportement des classifieurs lorsque les données sont affectées par des perturbations. Nous sommes convaincus que nos résultats seront utiles afin d'évaluer objectivement la fiabilité des classifieurs et construire des systèmes plus robustes.

Mots clefs : classification, robustesse, perturbations adverses, bruit aléatoire, bruit semi-aléatoire, invariance, transformations géométriques, nuisances, occlusions, apprentissage profond, réseaux neuronal convolutif.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

The goal of many computer vision tasks is to estimate categorical properties from visual data, such as the presence or absence of a particular object in a scene. For example, in the context of self-driving vehicles, one of the key tasks is to accurately recognize cars, traffic signs and pedestrians, even when these are affected by clutter, noise, occlusions and other forms of perturbation. One of the major difficulties in classification tasks is to correctly recognize an object despite the large amount of variability and corruptions that might affect the visual data in real-world tasks. While the human visual system is partially robust to such perturbations, it is unclear whether classifiers enjoy the same robustness properties. Understanding the robustness of classifiers to real-world perturbations is therefore primordial in the quest of designing effective classifiers.

In this thesis, our primary focus is the analysis and the quantification of the classifiers' robustness to different perturbations in the data. We study a diverse set of perturbations, ranging from *adversarial noise* to *random noise*, as well as more *structured* nuisances such as geometric transformations and occlusions. These different perturbation models are likely to affect the data in a widespread set of applications. First, a classifier might be subject to *adversarial* attacks, where a malicious agent having (partial or full) knowledge of the classification model *minimally* alters the samples so as to "fool" the classifier. For example, in person identification, an adversary would seek to minimally modify an existing photo/fingerprint to fool the classifier to be granted access in restricted areas. In such hostile environments, the robustness of classifiers against adversarial perturbations is therefore crucial. In other classification applications, the captured data might be subject to *random* noise due to the sensing process, data transmission, or any other component of the data acquisition pipeline. This noise regime is very different from the adversarial regime, as no prior knowledge on the classifier is used in the corruption process. Moreover, in computer vision applications, visual data often undergoes *structured* perturbations, such as geometric transformations, illumination changes and occlusions. These structured perturbations can be seen as some sort of data corruption that act on an ideal representation of the object. In all these cases, the major challenge in classification problems is then to *factor out* such corruptions or nuisances, and recover the categorical property of interest from the perturbed versions of the data.

The importance of analyzing the robustness of classifiers to different perturbations in the data (e.g., adversarial perturbations) goes well beyond the designated applications (e.g.,

security-related problems). In fact, the quantification of the amount of noise that is required to modify the estimated label of an image is crucial for understanding the concepts that are learned by the classifier. Perturbations reveal indeed the differences between the classes from the point of view of the classifier, and therefore provide important insights on the topology of the decision boundaries that separate the classes. For example, the analysis of the perturbation required to transform a (typical) "car" image into an image classified as "plane" allows us to understand the difference between these two concepts from the perspective of the classifier. While such a perturbation would typically include specific semantic objects (such as "airplane wings") for a human observer, it is unclear whether the typical classifiers use similar cues. The analysis of the robustness properties of classifiers is therefore crucial for improving our understanding of classifiers that are often seen as black box models.

Despite the importance of robust classification, many questions pertaining to the robustness of classifiers for different perturbation models remain open. In particular, while recent works (e.g., [Sze+14]) showed that state-of-the-art classifiers are extremely unstable to adversarial perturbations, the causes of instability remain unclear. Moreover, the effect of other *typical* perturbations on classifiers is unknown. For example, how are state-of-the-art deep neural networks affected by random or partially random noise? We believe these problems require quantitative answers and in-depth analyses, as they determine the reliability of such classifiers in real-world environments where perturbations occur almost systematically. An even more fundamental question is whether it is actually *feasible* to learn robust *and* accurate classifiers. More generally, it is crucial to study the interplay between the accuracy of a classifier and its robustness. Finally, when images undergo *structured* perturbations (such as geometric transformations), it is important to *quantify* the robustness of a classifier to such transformations. While most research papers report the accuracy of the classifier on a standard split of training and testing sets, this measure says little about the robustness of this classifier to structured nuisances. Designing an appropriate measure that summarizes the robustness of a classifier to structured nuisances (such as geometric transformations, or occlusions) is therefore an important question. In particular, it permits to understand the weaknesses of classifiers with respect to real-world nuisances.

## 1.1 Thesis outline

The thesis is organized as follows:

In Chapter 2, we review relevant works from the literature that relate to image classification and in particular robustness and invariance properties.

In Chapter 3, we study the problem of *estimating* the robustness of classifiers to adversarial perturbations. The estimation of the classifier's robustness involves the minimization of a non-convex objective function. We propose an efficient and accurate algorithm for estimating the robustness of classifiers that is based on an iterative linearization of the classifier's decision function. The proposed method is shown experimentally to compare favorably with respect to other methods in terms of the robustness estimation. Experimental results moreover show that augmenting the training set with adversarial examples can

increase the robustness to adversarial perturbations, and act as a regularizer to improve the accuracy of the classifiers.

Next, in Chapter 4, we analyze theoretically the robustness of classifiers to adversarial perturbations. The goal of this chapter is specifically to quantify how large the robustness to adversarial perturbations can be for fixed classification families (e.g., the family of linear classifiers). To do so, we establish learning-independent upper bounds on the robustness to adversarial perturbations, and reveal the existence of a *tradeoff* between robustness and classification accuracy. Specifically, we show that, for common classification tasks, it is *not* possible to find a classifier in the considered family that achieves both a large robustness and a large accuracy, independently of the training algorithm used to choose the classifier. We precisely quantify this tradeoff using data-dependent measures that capture the difficulty of the classification task with respect to the classifiers' family. Our analysis moreover suggests that the lack of robustness of classifiers is due to the lack of flexibility of classifiers with regards to the difficulty of the classification task, and that the robustness increases with the flexibility of the classification model. Experimental results finally confirm the theoretical results.

In Chapter 5, we study the robustness of classifiers in a novel *semi-random noise regime*, which generalizes random and adversarial noise. In the random noise regime, data points are perturbed by noise the direction of which is picked uniformly at random in the input space. The semi-random regime generalizes this model to random subspaces of arbitrary dimension, where a worst-case perturbation is sought within the subspace. We conduct a unified theoretical analysis on the robustness of classifiers in this general noise regime, and establish precise bounds on the robustness of classifiers that depend on the *curvature* of the classifier's decision boundary. Specifically, we show that the robustness of classifiers to random noise behaves as $\sqrt{d}$ times the robustness to adversarial perturbations (where $d$ denotes the dimension of the data) provided the curvature of the decision boundary is sufficiently small. This result highlights the blessing of dimensionality for the robustness of classifiers to random noise, as it shows that it is theoretically possible to achieve a large robustness to random noise even if the classifier is largely unstable to adversarial perturbations. Our bounds notably extend to the general semi-random regime, where we show that the robustness precisely behaves as $\sqrt{d/m}$ times the distance to boundary, with $m$ the dimension of the subspace. This result shows in particular that, even when $m$ is chosen as a small fraction of the dimension $d$, it is still possible to find small perturbations that are constrained to be in the subspace of dimension $m$ and that cause data misclassification. We conclude the chapter with experimental results showing that our theoretical estimates are very accurately satisfied by state-of-the-art deep neural networks on various sets of data, which suggests that the curvature of the decision boundary of such classifiers is small.

The focus of Chapter 6 is to study and quantify the *invariance* of classifiers, that is, their robustness to geometric transformations of the images (e.g., translation, rotations, etc...). We propose a principled and systematic method to measure the robustness of arbitrary image classifiers to geometric transformations. Specifically, we define the invariance measure as the minimal distance between the identity transformation and a transformation that is sufficient to change the decision of the classifier. In order to define the transformation metric, the key idea is to represent the set of transformed images as an image manifold; the transformation metric is then naturally captured by the *geodesic* distance on the

manifold. We propose a numerical algorithm for estimating the robustness of arbitrary classifiers to low-dimensional transformation groups, which is based on the efficient Fast Marching algorithm for computing geodesic distances on manifolds. We conduct an in-depth experimental evaluation of the proposed metric and show that our method is able to quantify the gain in invariance due to the increase of the depth of a convolutional neural network, as well as the effect of data augmentation on the invariance of classifiers. The proposed tool is then used to compare different networks in terms of their invariance, and can readily be used to objectively assess the reliability of classifiers to geometric perturbations.

Finally, in Chapter 7, we generalize the idea of assessing the invariance of classifiers to arbitrary parametric nuisance factors, such as occlusions, illumination changes or complex geometric transformations. To do so, we develop a general probabilistic framework, whose outcome is two-fold: the *estimation* of the robustness of classifiers to arbitrary nuisances, and the efficient *sampling* from problematic regions in the nuisance space that potentially lead to misclassification. The latter sampling technique is used to gain insights into the "weak spots" of classifiers, as well as on the features used to distinguish between classes. We apply the proposed framework to evaluate the robustness of state-of-the-art classifiers in natural image classification and face recognition tasks, and show that these classifiers are often only mildly robust to standard nuisances, such as occlusions. The visualization of problematic samples moreover shows, for example, that a slight occlusion of specific features in a face image can cause important errors in classification. Hence, besides the possibility to measure the robustness of classifiers to standard nuisances, the proposed framework can also be used to gain further insights on the actual discriminative image features that are used in the classification process.

## 1.2 Summary of contributions

The main contributions of this thesis are summarized as follows:

- We propose an accurate and efficient algorithm for estimating the robustness of classifiers to adversarial perturbations.
- We analyze the robustness of classifiers to adversarial perturbations, and show the existence of learning-independent fundamental limits on the robustness of classifiers. These limits depend on the classification risk, and reveal a novel robustness-risk tradeoff.
- We show that, for classifiers with sufficiently flat decision boundaries, the robustness to random noise is larger than the robustness to adversarial noise by a factor of the square root of the data dimenstion.
- More generally, we analyze the robustness of classifiers to a noise model that interpolates between random noise and adversarial noise, where minimal perturbations are sought in a randomly chosen subspace. We derive precise bounds on the robustness of classifiers in this generalized noise regime in terms of the curvature of the decision boundary and the subspace dimension. We show that state-of-the-art classifiers remain vulnerable to noise that is mostly random, and only mildly adversarial (i.e., where the subspace dimension is small).

- We propose a novel invariance score that measures the robustness of a classifier to geometric transformations, and propose an algorithm for computing the invariance score.
- We finally introduce a new probabilistic framework for assessing the robustness of classifiers to parametric nuisance sets, such as occlusion or illumination changes. The proposed framework allows us to discover the "weak spots" of any given classifier using appropriate sampling strategies.

# 2 Prior art

## 2.1 Outline

In this chapter, we review the relevant works from the literature that are linked to the general problems addressed in this thesis. We first review in Section 2.2 some of the major advances in the problem of image classification, with a special emphasis on deep convolutional networks, as these have been recently shown to largely outperform other architectures. Next, in Section 2.3, we focus on the problem of robust classification. We then delve into the related problem of achieving geometric invariance to transformations and other nuisance variables in the data in Section 2.4. In this chapter, we give particular emphasis on works that analyze and provide a better understanding of the inner mechanisms of modern classification methods.

## 2.2 Advances in image classification

In this section, we review some of the key works in the broad domain of image classification, with a special focus on recent architectures. The main challenge for accurately solving a visual task (in particular, image classification) is building a successful *visual representation*, which defines a mapping from pixels to meaningful features that can be used to solve the task. Since the very beginning of computer vision, a large number of visual representations have been proposed to tackle visual tasks [Sze10]. Prior to the popularization of *deep learning* for image classification, visual representations were mostly built in a *hand-engineered* fashion. Examples of such representations are SIFT [Low04], HOG [DT05], SURF [Bay+08]. See [MS05] for other examples. These feature representations leverage the human expertise to define plausible mappings from the pixels to features that satisfy the required properties (such as local invariance to rotations and translation). Unlike hand-engineered representations, the modern approach builds *deep* visual representations by gradually transforming the image into more abstract (and more useful) representations using a number of elementary operations. Deep representations take a broad inspiration from the hierarchical nature of the architecture of the visual cortex, where visual data flows from the retina to different visual areas [TFT01]. Deep visual representations are moreover *learned* from the data and impose no specific priors related to the problem modality. Thanks to the rapid development of GPUs, it is now possible to learn huge deep networks (possibly containing hundreds of millions of parameters) that significantly outperform previous state-of-the-art results in

many problems, such as natural image classification (ImageNet) [KSH12; Sze+15; SZ14], face recognition [PVZ15; Tai+14], and fine-grained classification [Wah+11; JSZ+15]. Deep convolutional networks build a representation of the data through the composition of linear and nonlinear elementary operations. We briefly mention three of the key operations used in deep convolutional networks as follows:

- *Convolution*: Given a three-dimensional feature map $\boldsymbol{x} \in \mathbb{R}^{H \times W \times D}$, the convolution layer convolves $\boldsymbol{x}$ with learned filters $\boldsymbol{f} \in \mathbb{R}^{H' \times W' \times D \times D''}$, and outputs $\boldsymbol{y} \in \mathbb{R}^{H'' \times W'' \times D''}$; i.e.,

$$y_{i'',j'',d''} = b_{d''} + \sum_{i'=1}^{H'} \sum_{j'=1}^{W'} \sum_{d=1}^{D} f_{i',j',d,d''} x_{i''+i'-1,j''+j'-1,d},$$

  where $b_{d''}$ denotes the bias, and $H'' = 1 + H - H'$ and $W'' = 1 + W - W'$.

- *Rectification*: Modern deep convolutional neural networks use a half-rectification activation functions defined by

$$\boldsymbol{y} = \max(0, \boldsymbol{x}).$$

  This simple nonlinearity has been shown to provide significant improvements with respect to traditional sigmoid activation functions [GBB11; MHN13], and is also tightly related to sparse coding [FDF15].

- *Pooling*: The goal of a pooling operation is to provide invariance to the classifier, by computing summary statistics over groups of features. Given a feature map $\boldsymbol{x}$, the pooled representation is given by

$$y_{i'',j'',d} = P\left(\left\{x_{i''+i'-1,j''+j'-1,d}\right\}_{\substack{1 \le i' \le W' \\ 1 \le j' \le H'}}\right),$$

  where $W'$ and $H'$ denote respectively the width and heights of the pooling regions, and $P$ denotes the pooling operator. Successful pooling operators include the *average* and *maximum*; see e.g., [BPL10] for a comparison between different pooling mechanisms. A pooling operation is often accompanied with a subsampling of the feature map.

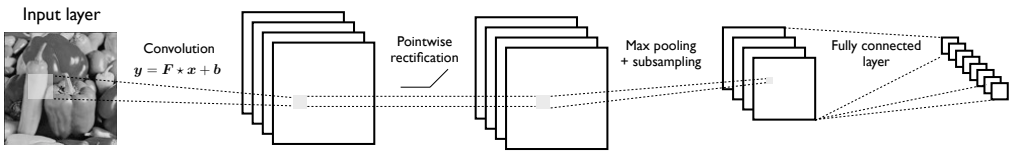



Figure 2.1: Structure of a CNN, obtained by stacking a series of linear and nonlinear elementary operations.

We refer to [VL15] for example for a detailed and hands-on description of other types of elementary operations commonly used in CNN architectures. The resulting classifier is a composition of many such operations, resulting in networks with possibly millions of

unknown parameters. An example architecture of a simple CNN is illustrated in Fig. 2.1. Once the architecture of the network is specified, the convolutional neural network is *trained* in an end-to-end fashion using example images. Specifically, this step involves learning the convolutional filters $\boldsymbol{f}$. Convolutional neural networks are generally trained using stochastic gradient descent (SGD) optimization, where the chain rule – or backpropagation [LeC+98b] – is used in order to compute the gradients. Alternative optimization algorithms have however been proposed recently, and have been shown to improve over standard optimization algorithms [KB14; MG15; Mar10].

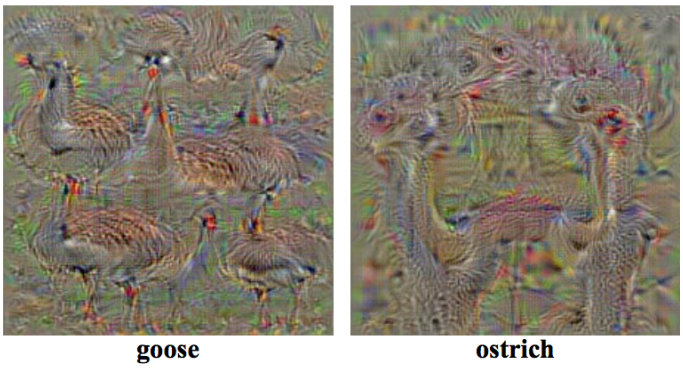



Figure 2.2: Images obtained using the visualization tool of [SVZ13] where the "goose" and "ostrich" neurons in the last layer of a deep neural network are maximized. Image taken from [SVZ13].

The recent impressive success of deep convolutional neural networks has triggered a significant number of fundamental research questions related to our understanding of the internal mechanisms leading to these successes. In an attempt to understand the features learned by deep networks, visualization strategies have been proposed in [MV15; DB16]. These visualization tools study the *invertibility* of the CNN representations; that is, to which extent an image $\boldsymbol{x}$ can be recovered from its feature representation $\phi(\boldsymbol{x})$? The visualization of these inverse images gives an understanding of the information that is preserved in the feature representations. In particular, these visualization strategies provide some empirical understanding about the layers at which invariance in the representation is achieved. Other visualization tools have been proposed in [ZF14; SVZ13], where the authors focus instead on the visualization of images that maximize the activation of single neurons. Fig. 2.2 shows images obtained by maximizing class-specific neurons in a state-of-the-art deep neural network. Experiments with these visualization tools show that, while neurons in lower layers mostly fire in the presence of edges, neurons in higher layers tend to be more sensitive to semantic objects with similar visual appearance. The invertibility of convolutional networks has also been studied from a theoretical perspective. In [BSL14], the invertibility of pooling operations used in neural network representations is analyzed. In particular, it is shown that, when the linear weights of the neural network are sufficiently redundant, it is possible to recover the original signals from pooled representations. Along the same lines, [ALM15] show that neural nets have an associated simple generative model that generates input data according to the conditional distribution characterizing the neural network.

Since the elementary operations that constitute convolutional neural networks are all linear or piecewise linear, the associated feature mapping is globally piecewise linear in

the image space. In [Mon+14], the number of linear regions (in the input space) of deep neural networks is studied; this number is shown to grow exponentially with the number of layers and polynomially in the size of the hidden layer. This growth therefore highlights that deep networks are more flexible than shallow ones, and hence can compute more "complicated" functions, which partly explains their success. Along the same lines of a theoretical understanding of the effect of deep networks on the input space, the authors in [ABB15] study the effect of applying half-rectification non-linearities on the input space, and in particular, on the linear separability of the datapoints.

On the optimization front, several works have attempted to understand the landscape of the non-convex objective function used to train deep neural networks [Dau+14; Cho+14]. In [Dau+14], the authors show that the difficulty in training such neural networks comes from the abundance of saddle points, not local minima, particularly in high dimensional problems that we typically encounter in neural network training. The authors empirically show that such saddle points are surrounded by high error plateaus that can make learning more difficult. In [Cho+14], the authors provide a theoretical description of the optimization of large neural networks, under strong statistical assumptions on the network. It is shown in particular that, under such assumptions, most local minima of the objective function are equivalent in the sense that they yield similar performance on a test set. The probability of finding a bad local minima (i.e., large objective function) is further shown to decrease with the network size; i.e., while for small-sized networks, "bad" local minimas have a non-zero probability of being recovered, the probability of recovering such local minimas for large networks is actually very small. Despite the strong statistical assumptions imposed on the network, this work provides a better understanding of the shape of the objective function, and explain why it is possible to train very deep networks using simple algorithms such as stochastic gradient descent.

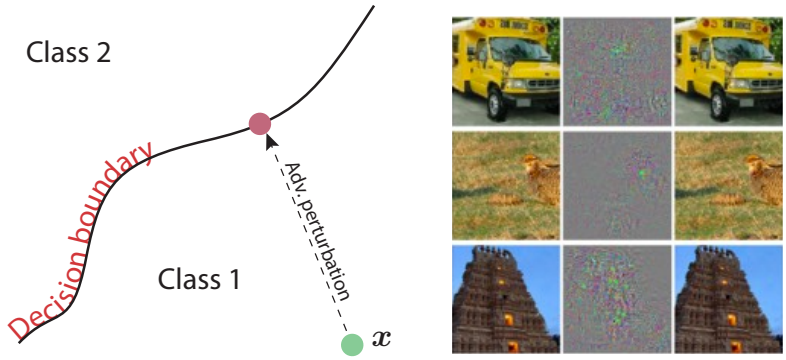
One of the main goals of this thesis is to develop analytical results in order to increase our understanding of classification models, and in particular, the *robustness of classifiers* to perturbations in the data. In the following sections, we review recent works related to the robustness of classifiers to different perturbation models.

## 2.3 Classification robustness

#### Robustness of neural networks to adversarial perturbations

State-of-the-art deep neural networks have recently been shown to be unstable to *adversarial perturbations* in the data [Sze+14]. Unlike random noise, adversarial perturbations are *minimal* (or *worst-case*) perturbations that are sought to change the estimated label of the classifier. The computation of adversarial perturbations involves solving an optimization problem (that requires the knowledge of the classifier's model), with the goal of going beyond decision boundary of the classifier (see Fig. 2.3 (a) for an illustration). On vision tasks, the results of [Sze+14] have shown that perturbations that are hardly perceptible to the human eye are sufficient to change the decision value of a deep network, even if the classifier has a performance that is close to the human visual system (Fig. 2.3 (b)). This surprising instability to "invisible" perturbations has received a widespread interest, as it highlights a fundamental difference with the human visual system, and raises important

challenges.




(a) (b)

Figure 2.3: Illustration of adversarial perturbations. **Left:** schematic representation of the perturbation. **Right:** Example images and adversarial perturbations. The left column depict original images (correctly classified by the network), the middle column shows the perturbations, and the right column shows the perturbed images (original image + perturbation) that are wrongly classified. This figure is taken from [Sze+14].

Beyond the obvious security concerns that this instability issue raises when classifiers are deployed in hostile environments, it also reveals fundamental shortcomings in the concepts and decision boundaries learned by state-of-the art classifiers. Specifically, this instability shows that most datapoints lie very closely to decision boundaries, even for two classes with significantly different semantic meaning. It should be noted that the authors in [Sze+14] have previously imputed the high instability of deep neural networks to the high "nonlinearity" of these classifiers. More precisely, the large Lipschitz constants associated to the feature map of the CNN are thought to induce "blind spots" in the classifier causing the instability to small perturbations. This explanation does not, however, take into account the important fact that linear classifiers are not exempt from this instability to adversarial perturbations. In that context, one of the contributions of this thesis is to provide a quantitative analysis of the robustness of classifiers to adversarial perturbations, with the goal of gaining a better understanding of this phenomenon.

While in [Sze+14], it is assumed that the adversary has full knowledge of the classification model, other works have examined the robustness of classifiers to perturbations when the adversary has only limited knowledge [Big+13; DSX10; GR06]. For example, in [Big+13], the transformed data point is constrained to remain within a maximum distance from the original sample, and it is further assumed that the attacker does not have direct access to the model (but only to a surrogate). These attack models are more likely to occur in real-world applications, as attackers often do not have complete control over the classification model.

#### Construction of robust classifiers with robust optimization

Following the original paper [Sze+14], several attempts have been made to construct deep networks with better robustness to adversarial perturbations [CPE14; GR14]. While most of these recent works that attempt to construct robust classifiers have mostly focused on the robustness of the new generation of neural networks to perturbations in the data, the

design of robust classifiers has been an active area of research and a long standing goal in machine learning. Using a robust optimization approach, new learning algorithms have been proposed for various learning tasks (see [CMX12] and references therein). A particular emphasis was given to the extension of support vector machines (SVMs) to settings where uncertainty corrupts the data. Assuming a disturbance model on the data samples, the robust optimization approach for constructing robust classifiers seeks to minimize the worst possible empirical error under such disturbances. In the context of SVM classification, the problem of learning robust linear classifiers $(\mathbf{w}, b)$ corresponds to the following min-max problem

$$\min_{\boldsymbol{w},b} \max_{\boldsymbol{U}\in\mathcal{U}} \left\{ r(\boldsymbol{w},b) + \sum_{i=1}^{m} \left(1 - y_i((\boldsymbol{x}_i + \boldsymbol{u}_i)^T \boldsymbol{w} + b)\right)_+ \right\},$$

where $\boldsymbol{x}_i$ and $\boldsymbol{u}_i$ denote respectively the datapoints and perturbation vectors, $y_i \in \{+1, -1\}$ denote the labels and $r(\boldsymbol{w}, b)$ is a regularization term. For certain uncertainty sets $\mathcal{U}$, the objective function can be written as a tractable convex optimization problem [XCM09; Lan+03; Bha04; TG07], which makes the task of finding a robust classifier feasible.

Other works have applied robust optimization approaches to design robust classifiers against new forms of perturbation. For example, in [GR06; DSX10], the authors propose a robust optimization algorithm to train classifiers that are robust against missing features (e.g., missing pixels in a handwritten recognition task). In [CM08], the problem of robust classification is examined when the noise affects the labels rather than the datapoints; it is then shown that robust classifiers can also be trained using robust optimization.

Unlike the above works that mostly propose robust classification algorithms for SVM, we focus in this thesis on more analytical aspects; e.g., *can* we actually find classifiers that are robust to perturbations? Moreover, we mostly concentrate in this thesis on modern successful architectures (e.g., deep neural networks) that have achieved huge progress in the problem of image classification. It should be noted that the construction of robust classification methods for deep neural networks is still an open problem.

### Robustness at the learning stage

While most of the above works consider attacks that alter datapoints at test time, it is equally important to achieve robustness to perturbations at the *training* stage. In particular, the adversary might manipulate the training data, therefore leading to a modification of the learned classification rule. This type of attack, dubbed *poisoning* attack, injects specially crafted training samples in order to maximize the test error of the learned classifier. The effect of poisoning attacks on different learning algorithms have previously been investigated in [BNL12; Xia+15]. In [Bar+06; Dal+04], a taxonomy of different attacks at the training stage requiring different levels of knowledge about the machine learning systems, and methods to counter these attacks are described. It is important to stress that, while these works study attacks that manipulate *the learning system* (e.g., change the decision function by injecting malicious training points), as well as defense strategies to counter these attacks, our focus in this thesis is more on robustness of fixed classifiers (not the learning algorithms). We finally note that the stability of learning algorithms has also been defined and studied

in [BE02; LP94a]. Specifically, in [BE02], the stability of learning algorithms is examined with respect to a removal of an element in the training set; this notion of stability is further shown to be useful in order to derive generalization error bounds. This is again a property of the learning algorithm; we are however more interested in this thesis on the robustness of fixed classifiers.

## 2.4 Classification invariance to geometric transformation and nuisance factors

In visual tasks, it is not only crucial to have classifiers that are robust against additive or adversarial attacks; it is also equally important to achieve invariance to structured nuisance variables, such as illumination changes, occlusions or standard local geometric transformations of the image. Specifically, when images undergo such structured deformations, it is desirable that the estimated label remains the same. In this part, we will first review some of the works that impose invariance by modifying the distance measure or by appropriate alignment. Then, we will review modern techniques that implicitly incorporate invariance in deep feature representation. Finally, we will review the relevant works in the literature that *assess and analyze* the invariance of classifiers to nuisance factors.

#### Transformation invariant distances and image alignment

One approach to introduce invariance in pattern recognition algorithms is to use transformation-invariant *distance measures*. The geometrically transformed versions of a fixed image span a low-dimensional nonlinear manifold in the high-dimensional space. Therefore, an appropriate invariant distance measure in this case corresponds to the manifold distance between these two transformation manifolds. Computing the transformation-invariant distance between two patterns or, equivalently, the manifold distance is unfortunately a difficult problem in general. The authors in [Sim+00] locally approximate the transformation-invariant distance with the distance between the linear spaces that are tangent to both manifolds. [VL05] go beyond the limitations of local invariance in tangent distance methods by embedding the tangent distance computation in a multiresolution framework. In [KF09], global invariance is achieved by approximating the original pattern with a linear combination of atoms from a parametric dictionary. Thanks to this approximation, the manifold is given in a closed form, and the objective function becomes equal to a difference of convex functions that can be globally minimized using cutting plane methods. Unfortunately, this class of optimization methods has a slow convergence rate with complexity limitations in practical settings.

Another approach for computing the transformation-invariant distance is to *align* (or *register*) the images. Transformation-invariant distances can then be easily computed from the aligned versions of the image. Feature-based approaches [Low04; DT05; Bay+08] represent an efficient class of methods for image registration. They are usually built on several steps: (i) *feature detection*, which searches for stable distinctive locations in the images, (ii) *feature description*, which provides a description of each detected location with an invariant descriptor, (iii) *features matching* between the images and (iv) *transformation estimation* that estimates the global transformation by looking at matched features. Note that it is crucial in this class of methods to describe the features in a transformation-

invariant way for easier matching. Many other approaches for image alignment exist [Pen+10; Mae+97; Ash07; FF13]; a comprehensive review of these approaches however goes outside the scope of this thesis, and we refer to [ZF03; Sze10] for surveys on this topic.

### Invariant feature representations

Deep convolutional networks build features that intend to be invariant to local geometric deformations in the data, through the use of a cascade of convolution, pooling and nonlinearity operators as discussed earlier in this chapter [Jar+09]. Despite the success of these architectures, their invariance properties are not fully understood. What is the effect of the number of layers on the invariance of the architecture? Which nonlinearity and pooling operations to use in order to enhance the invariance of the global representation? In [Mal12; BM13], the authors use a similar structure to convolutional neural networks (i.e., cascade of filtering, nonlinearity and pooling operations), and *impose* the requirement of stability of the representation to local deformations, while retaining maximum information about the original data. Formally, the feature mapping $\Phi$ is imposed to satisfy the following stability conditions:

$$\text{Stability to additive noise:} \quad \|\Phi\epsilon\| \leq C\|\epsilon\|, \tag{2.1}$$

$$\text{Stability to local deformations:} \quad \|\Phi x_\tau - \Phi x\| \leq C\|x\|\sup_u|\nabla\tau(u)|, \tag{2.2}$$

where $\tau(u)$ is a displacement field that deforms the image, $x_\tau(u) = x(u - \tau(u))$, and $|\cdot|$ is the matrix operator norm. It should be noted that the condition in Eq. (2.2) implies the invariance of the feature representation to global translations (i.e., special case where $\nabla\tau(u) = 0$). In order to satisfy these constraints, a new architecture is proposed, the *scattering network*, where successive filtering with wavelets and pointwise nonlinearities are applied. It should be noted that the approach used to build this scattering network significantly differs from traditional convolutional neural networks, as no learning of the filters is involved. Scattering networks have been shown to achieve very high classification accuracies on digit and texture classification tasks in [BM13; SM13]. Scattering networks have also been applied to more complex tasks, such as natural image classification in [OM15]. While yielding better results with respect to previously known unsupervised dictionary learning methods and fixed-feature classification methods, scattering networks still underperform supervised CNNs in terms of classification accuracy on these complex tasks. In another effort to improve the invariance properties of deep convolutional neural networks, the authors in [JSZ+15] proposed a new module, the spatial transformer, that *geometrically* transforms the filter maps. Spatial transformer modules, similarly to convolutional modules, are trained in a supervised way; in particular, the estimation of the transformation is performed in order to maximize the classification accuracy. Using spatial transformer networks, the performance of classifiers improve significantly, especially when images have noise and clutter, as these modules automatically learn to localize and unwarp corrupted images. Finally, another popular way of building more invariant representations is through virtual jittering (or data augmentation), where training data are transformed and fed back to the training set. One of the drawbacks of this approach is however that the training can become intractable, as the size of the training set becomes substantially larger than the original data set. To make the training more efficient with the augmented training sets, new techniques have been proposed based on the non-uniform sampling of the training data

[CJF16]. Besides, some works [Pau+14; Hau+16] have recently developed principled and automatic approaches for transforming images. Despite these major advances in building more robust and invariant representations, a thorough understanding and assessment of the invariance to general nuisance factors of these classifiers remains open.

### Analyzing the invariance of classifiers

We review in this section works that assess and analyze the invariance of classifiers to transformations in the data. Several empirical works have been introduced to assess the invariance of classifiers to geometric transformations in the data. In [Goo+09], the authors develop simple tests to assess the invariance of neural networks to transformations of the data. These tests consider controlled images of gratings, and measure the effect of applying simple geometric transformations to the probe image. In a more recent work, [LV15] study the *equivariance* property of image representations, that is the relation between the features of the transformed images and those of the original image. The invariance is a special case of equivariance where transformation has no effect. These experiments help understanding at which layer invariance to simple transformations such as vertical flips, scale or rotation is achieved. In [Bak+16], the view-point invariance of CNNs is analyzed. In particular, the authors study the evolution of the view-manifold (that contain the features of the different views of an object) with respect to the number of layers. It is shown that the information on the view of an object is preserved till the last but one layer of the CNN. In other words, CNNs preserve the structure of the view manifold, which supports the hypothesis that this manifold in higher layers is "untangled" in higher layers, rather than being "collapsed", which goes in the same direction of [DC07] for the human brain. In [KDS16], an empirical analysis of the ability of current CNNs to manage location and scale variability is performed. It is shown in particular that CNNs are not very effective in factoring out location and scale variability, despite the popular belief that the convolutional architecture and the local spatial pooling provides invariance to such representations.

In [SC16; SDK15; ARP16], a more theoretical perspective is taken to analyze the invariance of modern classification methods to nuisance variables. Specifically, in [SC16], the authors propose a new mathematical formalism of visual representations, and define the notion of *optimal* representation. The optimality of a representation essentially formalizes the intuitive definition of a representation that satisfies the property of invariance to nuisance variables, and retaining maximal information of the original images. Connections with existing representations (shallow and deep) are further shown. This work has suggested well-grounded modifications of existing architectures that led to significant improvements in the problem of feature correspondence in single-view and multi-view settings [DS15; Don+15].

Despite the importance of the invariance property of classifiers to nuisance variables, there exists no systematic method to test the invariance of classifiers to arbitrary nuisance variables in the data, up to our knowledge. We provide in this thesis tools to *quantify* the invariance of black-box classifiers to arbitrary nuisance variables.

## 2.5 Summary

We summarize the main points of this chapter, in the light of the contributions of this thesis and upcoming challenges:

- Recent works have shown that impressive performance can be reached with deep CNN classifiers. This has triggered an important number of works that attempt to explain the success of such architectures.
- One of the fundamental properties of classifiers is their robustness to perturbations in the data. The robustness of deep CNN classifiers has nevertheless only been very recently investigated empirically. One of the aims of this thesis is to provide analytical results on the robustness of classifiers, and to lay a theoretical foundation for a rigorous study of the robustness.
- The invariance of classifiers to nuisance variables has also been studied to a large extent by previous works. Despite the abundance of analytical works that try to provide a better understanding of the invariance properties of modern classifiers, no framework exists for systematically measuring and quantifying the invariance to geometric transformations, or more generally, the robustness to general nuisance variables.

# 3 Estimation of classifiers' robustness

## 3.1 Introduction

In this chapter, we propose an algorithm to estimate the robustness of classifiers to *adversarial* perturbations. Adversarial perturbations (also called *worst-case* noise) are *minimal* perturbations that are sought to switch the estimated label of the classifier. In vision tasks, the empirical results of [Sze+14] have shown that adversarial perturbations that are hardly perceptible to the human eye are sufficient to change the classification decision of a deep network, even if the classifier has a very good accuracy. While [Sze+14] estimated adversarial examples by solving a series of penalized optimization problems, this algorithm unfortunately does not scale up to large datasets, and therefore hinders the understanding of the robustness of state-of-the-art deep neural networks in general settings. The efficient and accurate computation of adversarial examples is moreover a key component for improving the robustness of classifiers. We therefore believe that an accurate and efficient method to compute the robustness of classifiers is a necessary starting point towards a better understanding of the limits of current architectures and to design robust classification methods.

Our new algorithm for estimating the robustness to adversarial perturbations is based on an iterative linearization of the classifier's decision function. When the decision boundaries are linear, we show that the robustness can be expressed in closed form and our algorithm iteratively uses such update rules for an accurate computation of the robustness. We perform an extensive experimental comparison, and show that 1) our method computes adversarial perturbations more reliably and efficiently than existing methods 2) augmenting the training set with adversarial examples increases the robustness to adversarial perturbations. We also show that using imprecise approaches for the computation of adversarial perturbations could lead to different and sometimes misleading conclusions about the robustness. Hence, the proposed method provides a better understanding of this intriguing phenomenon and of its influence factors.

The chapter is organized as follows: in Section 3.2, we introduce important definitions and notations on the robustness of classifiers. We then present our algorithm in the binary case

Part of this chapter has been published in [MDFF16].

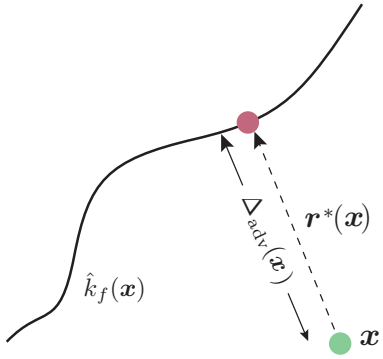



Figure 3.1: Schematic representation of an adversarial perturbation. The vector $\boldsymbol{r}^*(\boldsymbol{x})$ denotes the adversarial perturbation that moves the datapoint $\boldsymbol{x}$ to the boundary, and $\Delta_{\text{adv}}(\boldsymbol{x})$ denotes its $\ell_2$ norm.

in Section 3.3, and the multi-class case in Section 3.4. We finally provide experimental results in Section 3.5, where we compute the robustness of state-of-the-art classifiers, and compare the proposed optimization method for computing the robustness to other existing methods.

## 3.2 Definitions & notations

We first introduce the framework and notations that are used for analyzing the robustness of classifiers. Throughout this thesis, we use $f$ to denote an arbitrary classification function, and $\hat{k}_f$ the classification rule associated to $f$. In binary classification tasks, we have $f : \mathbb{R}^d \to \mathbb{R}$, and the estimated label of a datapoint $\boldsymbol{x} \in \mathbb{R}^d$ is typically obtained by taking the sign of $f(\boldsymbol{x})$; hence, $\hat{k}_f(\boldsymbol{x}) = \text{sign}(f(\boldsymbol{x}))$. For notation simplicity, when the classification function $f$ is clear from the context, we use $\hat{k}(\boldsymbol{x})$ instead in order to denote the estimated label of $\boldsymbol{x}$. Given a datapoint $\boldsymbol{x} \in \mathbb{R}^d$, we denote by $y(\boldsymbol{x})$ the *ground-truth label* of $\boldsymbol{x}$. We denote by $\mu$ the probability measure on $\mathbb{R}^d$ of the datapoints we wish to classify. We assume moreover that this probability distribution has bounded support; that is, $\mathop{\mathbb{P}}_{\boldsymbol{x}\sim\mu}(x \in B) = 1$, where $B = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \leq M\}$ for some $M > 0$. The performance of a classifier $f$ is usually measured by its *risk*, defined as the probability of misclassification measured on the data distribution $\mu$:

$$R(f) = \mathop{\mathbb{P}}_{\boldsymbol{x}\sim\mu}(\hat{k}(\boldsymbol{x}) \neq y(\boldsymbol{x})). \tag{3.1}$$

For a fixed datapoint $\boldsymbol{x} \in \mathbb{R}^d$, we define $\boldsymbol{r}^*(\boldsymbol{x})$ to be the *minimal* perturbation that changes the estimated label of the classifier[1] at $\boldsymbol{x}$; i.e.,

$$\boldsymbol{r}^*(\boldsymbol{x}) = \mathop{\text{argmin}}_{\boldsymbol{r}\in\mathbb{R}^d} \|\boldsymbol{r}\|_2 \text{ subject to } \hat{k}(\boldsymbol{x}+\boldsymbol{r}) \neq \hat{k}(\boldsymbol{x}). \tag{3.2}$$

[1] A borderline perturbation vector sending the datapoint exactly to the boundary is assumed to change the label of the classifier.

| Quantity | Notation | Dependence |
|---|---|---|
| Risk | $R(f)$ [Eq. (3.1)] | $\mu, y, f$ |
| Pointwise robustness to adversarial perturbations | $\Delta_{\text{adv}}(\boldsymbol{x})$ [Eq. (3.3)] | $f, \boldsymbol{x}$ |
| Average robustness to adversarial perturbations | $\rho_{\text{adv}}(f)$ [Eq. (3.4)] | $f, \mu$ |

Table 3.1: Quantities of interest and their dependencies.

The robustness to adversarial perturbations is quantified by taking the norm of the vector $\boldsymbol{r}^*(\boldsymbol{x})$:

$$\Delta_{\text{adv}}(\boldsymbol{x}) = \|\boldsymbol{r}^*(\boldsymbol{x})\|_2. \tag{3.3}$$

In the above definitions of $\boldsymbol{r}^*(\boldsymbol{x})$ and $\Delta_{\text{adv}}(\boldsymbol{x})$, we removed the explicit dependence on the classifier $f$ to simplify notations, as the classifier will be clear from the context. An illustration of these quantities is provided in Fig. 3.1.

Note that unlike random noise, the above definition corresponds to a minimal noise, where the perturbation $\boldsymbol{r}$ is sought to flip the estimated label of $\boldsymbol{x}$. This justifies the *adversarial* nature of the perturbation. It is important to note that, while $\boldsymbol{x}$ is a datapoint sampled according to $\mu$, the perturbed point $\boldsymbol{x} + \boldsymbol{r}^*(\boldsymbol{x})$ is not required to belong to the dataset (i.e., $\boldsymbol{x} + \boldsymbol{r}^*(\boldsymbol{x})$ can be outside the support of $\mu$). It should also be noted that while we use here the $\ell_2$ norm to quantify the perturbation, other norm choices are also possible; we refer to Appendix A.2 for a discussion of the norm choice.

The global robustness to adversarial perturbation of $f$ is finally defined as the average of $\Delta_{\text{adv}}(\boldsymbol{x})$ over all $\boldsymbol{x} \sim \mu$:

$$\rho_{\text{adv}}(f) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mu}(\Delta_{\text{adv}}(\boldsymbol{x})). \tag{3.4}$$

In words, $\rho_{\text{adv}}(f)$ is defined as the average norm of the minimal perturbations required to change the estimated labels of the datapoints. Note that $\rho_{\text{adv}}(f)$ is a property of both the classifier $f$ and the distribution $\mu$, but it is independent of the true labels of the datapoints $y$.[2] Moreover, it should be noted that $\rho_{\text{adv}}$ is different from the margin considered by SVMs. In fact, SVM margins are traditionally defined as the *minimal* distance to the (linear) boundary over all training points, while $\rho_{\text{adv}}$ is defined as the *average* distance to the boundary over all training points. In addition, distances in our case are measured in the input space, while the margin is defined in the feature space for kernel SVMs. Table 3.1 shows a summary of the different quantities.

## 3.3 Computation of the robustness for binary classifiers

As a multiclass classifier can be viewed as aggregation of binary classifiers, we first propose an algorithm for measuring the robustness of binary classifiers. That is, we assume here $\hat{k}(\boldsymbol{x}) = \text{sign}(f(\boldsymbol{x}))$, where $f$ is an arbitrary scalar-valued image classification function $f : \mathbb{R}^d \to \mathbb{R}$. We also denote by $\mathscr{B} \triangleq \{\boldsymbol{x} : f(\boldsymbol{x}) = 0\}$ the zero level set of $f$, which represents

[2] In that aspect, our definition slightly differs from the one proposed in [Sze+14], which defines the robustness to adversarial perturbations as the average of the norms of the minimal perturbations required to *misclassify* all datapoints.

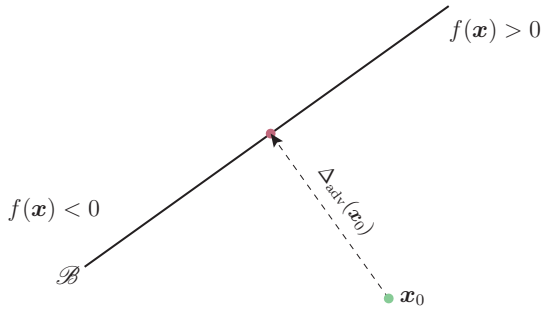



Figure 3.2: Adversarial examples for a linear binary classifier.

the classifier's *decision boundary*. We begin by analyzing the case where $f$ is an affine classifier $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$, and then derive the general algorithm, which can be applied to any differentiable binary classifier $f$.

In the case where the classifier $f$ is affine, it can easily be seen that the robustness of $f$ at point $\boldsymbol{x}_0$, $\Delta_{\text{adv}}(\boldsymbol{x}_0)$, is equal to the distance from $\boldsymbol{x}_0$ to the separating affine hyperplane $\mathscr{B} = \{\boldsymbol{x} : \boldsymbol{w}^T\boldsymbol{x} + b = 0\}$ (Figure 3.2). The minimal perturbation to change the classifier's decision corresponds to the orthogonal projection of $\boldsymbol{x}_0$ onto $\mathscr{B}$. It is given by the closed-form formula:

$$\begin{aligned} \boldsymbol{r}^*(\boldsymbol{x}_0) &:= \operatorname{argmin} \|\boldsymbol{r}\|_2 \\ &\quad \text{subject to } f(\boldsymbol{x}_0 + \boldsymbol{r}) f(\boldsymbol{x}_0) \leq 0 \\ &= -\frac{f(\boldsymbol{x}_0)}{\|\boldsymbol{w}\|_2^2}\boldsymbol{w}. \end{aligned} \tag{3.5}$$

Assuming now that $f$ is a general binary differentiable classifier, we adopt an iterative procedure to estimate the robustness $\Delta_{\text{adv}}(\boldsymbol{x}_0)$ and the corresponding perturbation $\boldsymbol{r}^*(\boldsymbol{x}_0)$. Specifically, at each iteration, $f$ is linearized around the current point $\boldsymbol{x}_i$ and the minimal perturbation of the linearized classifier is computed as

$$\operatorname*{argmin}_{\boldsymbol{r}_i} \|\boldsymbol{r}_i\|_2 \text{ subject to } f(\boldsymbol{x}_i) + \nabla f(\boldsymbol{x}_i)^T \boldsymbol{r}_i = 0. \tag{3.6}$$

The perturbation $\boldsymbol{r}_i$ at iteration $i$ of the algorithm is computed using the closed form solution in Eq. (3.5), and the point $\boldsymbol{x}_{i+1}$ for the next iteration is computed as $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{r}_i$. The algorithm stops when $\boldsymbol{x}_i$ changes the sign of the classifier. The proposed algorithm for binary classifiers is summarized in Algorithm 1 and a geometric illustration of the method is shown in Figure 3.3.

In practice, Algorithm 1 can often converge to a point exactly on the zero level set $\mathscr{B}$. In order to reach the other side of the classification boundary, the final perturbation vector $\hat{\boldsymbol{r}}$ is multiplied by a constant $1 + \eta$, with $\eta \ll 1$. In our experiments, we have used $\eta = 0.02$.

**Algorithm 1** Computing minimal perturbation for binary classifiers

**input:** Image $\boldsymbol{x}$, classifier $f$.
**output:** Perturbation $\hat{\boldsymbol{r}}$.
Initialize $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$, $i \leftarrow 0$.
**while** $\operatorname{sign}(f(\boldsymbol{x}_i)) = \operatorname{sign}(f(\boldsymbol{x}_0))$ **do**
    $\boldsymbol{r}_i \leftarrow -\frac{f(\boldsymbol{x}_i)}{\|\nabla f(\boldsymbol{x}_i)\|_2^2} \nabla f(\boldsymbol{x}_i)$,
    $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \boldsymbol{r}_i$,
    $i \leftarrow i + 1$.
**end while**
**return** $\hat{\boldsymbol{r}} = \sum_i \boldsymbol{r}_i$.

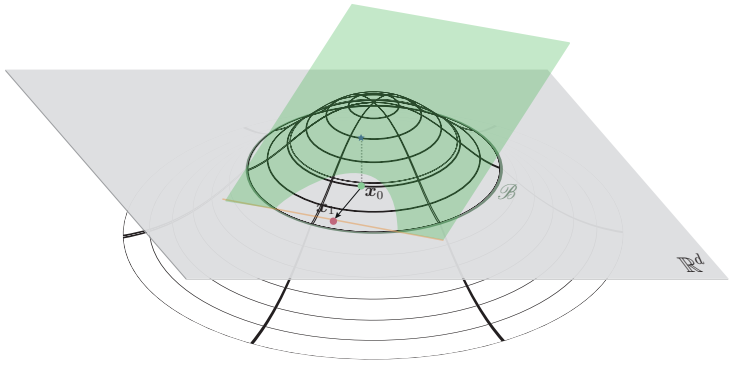



Figure 3.3: Illustration of Algorithm 1 for $d = 2$. Assume $\boldsymbol{x}_0 \in \mathbb{R}^d$. The green plane is the graph of $\boldsymbol{x} \mapsto f(\boldsymbol{x}_0) + \nabla f(\boldsymbol{x}_0)^T(\boldsymbol{x} - \boldsymbol{x}_0)$, which is tangent to the classifier function (wire-framed graph) $\boldsymbol{x} \mapsto f(\boldsymbol{x})$. The orange line indicates where $f(\boldsymbol{x}_0) + \nabla f(\boldsymbol{x}_0)^T(\boldsymbol{x} - \boldsymbol{x}_0) = 0$. $\boldsymbol{x}_1$ is obtained from $\boldsymbol{x}_0$ by projecting $\boldsymbol{x}_0$ on the orange hyperplane of $\mathbb{R}^d$.

## 3.4 Computation of the robustness for multiclass classifiers

We now extend our algorithm to the multiclass case, in particular to the commonly used one-vs-all classification schemes. In this scheme, the classifier has $L$ outputs where $L$ is the number of classes. Therefore, a classifier can be defined as $f : \mathbb{R}^d \to \mathbb{R}^L$ and the classification is performed as follows:

$$\hat{k}(\boldsymbol{x}) = \operatorname*{argmax}_k f_k(\boldsymbol{x}), \tag{3.7}$$

where $f_k(\boldsymbol{x})$ is the output of $f(\boldsymbol{x})$ that corresponds to the $k^{\text{th}}$ class. Similarly to the binary case, we first present our robustness estimation algorithm for the linear case and then we generalize it to other classifiers.

### 3.4.1 Affine multiclass classifier

Let $f(\boldsymbol{x})$ be an affine classifier, i.e., $f(\boldsymbol{x}) = \mathbf{W}^\top \boldsymbol{x} + \boldsymbol{b}$ for a given $\mathbf{W} \in \mathbb{R}^{d \times L}$ and $\boldsymbol{b} \in \mathbb{R}^L$. Since the mapping $\hat{k}$ is the outcome of a one-vs-all classification scheme, the minimal

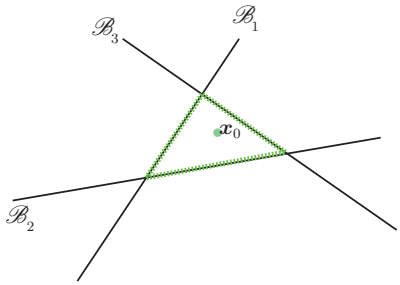



Figure 3.4: For $\boldsymbol{x}_0$ belonging to class 4, let $\mathscr{B}_k = \{\boldsymbol{x} : f_k(\boldsymbol{x}) - f_4(\boldsymbol{x}) = 0\}$ for $k = \{1, 2, 3\}$, denote the decision boundaries with respectively class $1, 2$ and 3. These hyperplanes are depicted in solid black lines and the boundary of polyhedron $P$ is shown in green dotted line. We recall that $P$ is the polyhedron defining the region where $f$ outputs label $\hat{k}(\boldsymbol{x}_0)$ ($\hat{k}(\boldsymbol{x}_0) = 4$ in this example).

perturbation $\boldsymbol{r}^*(\boldsymbol{x}_0)$ to fool the classifier at $\boldsymbol{x}_0$ can be rewritten as follows

$$\begin{aligned} \boldsymbol{r}^*(\boldsymbol{x}_0) = &\operatorname*{argmin}_{\boldsymbol{r}} \|\boldsymbol{r}\|_2 \\ &\text{s.t. } \exists k \neq \hat{k}(\boldsymbol{x}_0) : \boldsymbol{w}_k^\top(\boldsymbol{x}_0 + \boldsymbol{r}) + b_k \geq \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}^\top(\boldsymbol{x}_0 + \boldsymbol{r}) + b_{\hat{k}(\boldsymbol{x}_0)}, \end{aligned} \tag{3.8}$$

where $\boldsymbol{w}_k$ is the $k^{\text{th}}$ column of $\mathbf{W}$. Geometrically, the above problem corresponds to the computation of the distance between $\boldsymbol{x}_0$ and the *boundary* of the convex polyhedron $P$,

$$P = \bigcap_{k=1}^{L} \{\boldsymbol{x} : f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}) \geq f_k(\boldsymbol{x})\}, \tag{3.9}$$

where $\boldsymbol{x}_0$ is located inside $P$. The interior of polyhedron $P$ defines the region of the space where $f$ outputs the label $\hat{k}(\boldsymbol{x}_0)$. This setting is depicted in Figure 3.4. The solution to the problem in Eq. (3.8) is given as follows:

**Fact 1.** *Define $\boldsymbol{r}^k(\boldsymbol{x}_0)$ to be the optimal perturbation when only classes $k$ and $\hat{k}(\boldsymbol{x}_0)$ are considered:*

$$\boldsymbol{r}^k(\boldsymbol{x}_0) = \operatorname*{argmin}_{\boldsymbol{r}} \|\boldsymbol{r}\|_2 \text{ such that } f_k(\boldsymbol{x}_0 + \boldsymbol{r}) \geq f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0 + \boldsymbol{r}). \tag{3.10}$$

*When the functions $f_k$ are affine, we have for all $k \neq \hat{k}(\boldsymbol{x}_0)$,*

$$\boldsymbol{r}^k(\boldsymbol{x}_0) = \frac{|f_k(\boldsymbol{x}_0) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0)|}{\|\boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}\|_2^2}(\boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}). \tag{3.11}$$

*Moreover, the minimum perturbation $\boldsymbol{r}^*(\boldsymbol{x}_0)$ corresponds to the perturbation $\boldsymbol{r}^k(\boldsymbol{x}_0)$ with minimum $\ell_2$ norm.*

*Proof.* Let $\boldsymbol{x}_0 \in \mathbb{R}^d$. We have

$$\boldsymbol{r}^k(\boldsymbol{x}_0) = \operatorname*{argmin}_{\boldsymbol{r}} \|\boldsymbol{r}\|_2 \text{ such that } (\boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)})^T(\boldsymbol{x}_0 + \boldsymbol{r}) + b_k - b_{\hat{k}(\boldsymbol{x}_0)} \geq 0.$$

Note that, by definition of $\hat{k}(\boldsymbol{x}_0)$, we have $f_k(\boldsymbol{x}_0) \leq f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0)$. Hence, the above problem

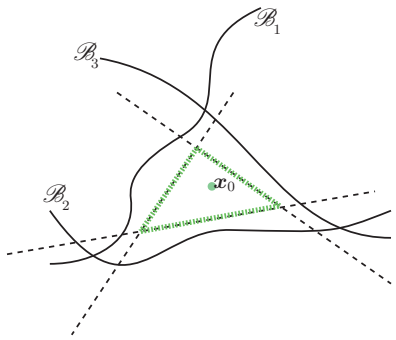



Figure 3.5: For $\boldsymbol{x}_0$ belonging to class 4, let $\mathscr{B}_k = \{\boldsymbol{x} : f_k(\boldsymbol{x}) - f_4(\boldsymbol{x}) = 0\}$ for $k \in \{1, 2, 3\}$ denote the decision boundaries with class 1, 2 and 3 respectively. We approximate these decision boundaries with affine hyperplanes, and the resulting decision boundary (that is the boundary of polyhedron $\tilde{P}_0$) is shown in green.

corresponds to the orthogonal projection of the datapoint $\boldsymbol{x}_0$ onto the affine hyperplane of $\mathbb{R}^d$ with normal vector $\boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}$ and bias $b_k - b_{\hat{k}(\boldsymbol{x}_0)}$. The orthogonal projection is given by the formula in Eq. (3.11). To relate $\boldsymbol{r}^*(\boldsymbol{x}_0)$ to $\boldsymbol{r}^k(\boldsymbol{x}_0)$, note that

$$\begin{aligned}
\|\boldsymbol{r}^*(\boldsymbol{x}_0)\|_2 &= \min_{\boldsymbol{r}} \|\boldsymbol{r}\|_2 \text{ s.t. } \boldsymbol{r} \in \bigcup_{k \neq \hat{k}(\boldsymbol{x}_0)} \{f_k(\boldsymbol{x}_0 + \boldsymbol{r}) \geq f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0 + \boldsymbol{r})\} \\
&= \min_{k \neq \hat{k}(\boldsymbol{x}_0)} \min_{\boldsymbol{r}} \|\boldsymbol{r}\|_2 \text{ s.t. } f_k(\boldsymbol{x}_0 + \boldsymbol{r}) \geq f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0 + \boldsymbol{r}) \\
&= \min_{k \neq \hat{k}(\boldsymbol{x}_0)} \|\boldsymbol{r}^k(\boldsymbol{x}_0)\|_2.
\end{aligned}$$

□

In other words, to compute $\boldsymbol{r}^*(\boldsymbol{x}_0)$, we project $\boldsymbol{x}_0$ on the different hyperplanes that form the polyhedron $P$, and we then choose the one with minimal $\ell_2$ norm. We recall that the robustness $\Delta_{\text{adv}}(\boldsymbol{x}_0)$ at $\boldsymbol{x}_0$ is then obtained by taking the $\ell_2$ norm of the perturbations; i.e., $\Delta_{\text{adv}}(\boldsymbol{x}_0) = \|\boldsymbol{r}^*(\boldsymbol{x}_0)\|_2$. The global robustness score $\rho_{\text{adv}}(f)$ is then estimated by computing the mean of $\Delta_{\text{adv}}(\boldsymbol{x})$ on datapoints $\boldsymbol{x} \sim \mu$.

### 3.4.2 General classifier

We now extend the algorithm to the general case of multiclass differentiable classifiers. For general non-linear classifiers, the set $P$ in Eq. (3.9) that describes the region of the space where the classifier outputs label $\hat{k}(\boldsymbol{x}_0)$ is no longer a polyhedron. Following the explained iterative linearization procedure in the binary case, we approximate the set $P$ at iteration $i$ by a polyhedron $\tilde{P}_i$

$$\begin{aligned}
\tilde{P}_i = \bigcap_{k=1}^{L} \Big\{ \boldsymbol{x} : f_k(\boldsymbol{x}_i) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i) \\
+ \nabla f_k(\boldsymbol{x}_i)^\top (\boldsymbol{x} - \boldsymbol{x}_i) - \nabla f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)^\top (\boldsymbol{x} - \boldsymbol{x}_i) \leq 0 \Big\}.
\end{aligned} \tag{3.12}$$

We then approximate, at iteration $i$, the distance between $\boldsymbol{x}_i$ and the boundary of $P$ by the distance to the boundary of $\tilde{P}_i$. Specifically, at each iteration of the algorithm, the perturbation vector that reaches the boundary of the polyhedron $\tilde{P}_i$ is computed, and the current estimate is updated. A schematic representation of the linearization of the decision boundary is shown in Fig. 3.5. The method is given in Algorithm 2. It should be noted that the proposed algorithm operates in a greedy way and is not guaranteed to converge to the optimal perturbation in Eq. (3.3). However, we have observed in practice that our algorithm yields very small perturbations that are believed to be good approximations of the minimal perturbation.

**Algorithm 2** Computing minimal perturbation in multi-class case

**input:** Image $\boldsymbol{x}$, classifier $f$.
**output:** Perturbation $\hat{\boldsymbol{r}}$.
Initialize $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$, $i \leftarrow 0$.
**while** $\hat{k}(\boldsymbol{x}_i) = \hat{k}(\boldsymbol{x}_0)$ **do**
    **for** $k \neq \hat{k}(\boldsymbol{x}_0)$ **do**
        Compute

$$\boldsymbol{r}^k(\boldsymbol{x}_i) = \frac{|f_k(\boldsymbol{x}_i) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)|}{\|\nabla f_k(\boldsymbol{x}_i) - \nabla f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)\|_2^2}(\nabla f_k(\boldsymbol{x}_i) - \nabla f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)). \tag{3.13}$$

    **end for**
    Let $\boldsymbol{r}_i$ be the vector $\boldsymbol{r}^k(\boldsymbol{x}_i)$ with minimal $\ell_2$ norm.
    $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \boldsymbol{r}_i$
    $i \leftarrow i + 1$
**end while**
**return** $\hat{\boldsymbol{r}} = \sum_i \boldsymbol{r}_i$

It should be noted that the optimization strategy employed to compute the perturbations is strongly tied to existing optimization techniques. In the binary case, it can be seen as Newton's iterative algorithm for finding roots of a nonlinear system of equations in the underdetermined case [Rus06]. This algorithm is known as the normal flow method. The convergence analysis of this optimization technique can be found for example in [WW90]. Our algorithm in the binary case can alternatively be seen as a gradient descent algorithm with an adaptive step size that is automatically chosen at each iteration. Finally, the linearization in Algorithm 2 is also similar to a sequential convex programming algorithm where the constraints are linearized at each step.

### 3.4.3 Extension to $\ell_p$ norm

While we have measured the perturbations using the $\ell_2$ norm, our algorithm for measuring the robustness is however not limited to this choice, and the proposed algorithm can simply be adapted to find minimal adversarial perturbations for any $\ell_p$ norm ($p \in [1, \infty)$). To do so, the update step in Eq. (3.13) of Algorithm 2 must be substituted by the following update

$$\boldsymbol{r}^k(\boldsymbol{x}_i) = \frac{|f_k(\boldsymbol{x}_i) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)|}{\|\boldsymbol{w}'_k\|_q^q}|\boldsymbol{w}'_k|^{q-1} \odot \operatorname{sign}(\boldsymbol{w}'_k),$$

with $\boldsymbol{w}'_k = \nabla f_k(\boldsymbol{x}_i) - \nabla f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)$, and where $\odot$ is the pointwise product and $q = \frac{p}{p-1}$.[3] In particular, when $p = \infty$ (i.e., the supremum norm $\ell_\infty$), this update step becomes

$$\boldsymbol{r}^k(\boldsymbol{x}_i) = \frac{|f_k(\boldsymbol{x}_i) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)|}{\|\boldsymbol{w}'_k\|_1} \operatorname{sign}(\boldsymbol{w}'_k). \tag{3.14}$$

## 3.5 Experimental results

### 3.5.1 Setup

We now test our algorithm on deep convolutional neural networks architectures applied to MNIST [LeC+98a], CIFAR-10 [KH09], and ImageNet [Rus+15] image classification datasets. We consider the following deep neural network architectures:

- **MNIST:** A two-layer fully connected network, and a two-layer LeNet convolutional neural network architecture [LeC+99]. Both networks are trained with SGD with momentum using the MatConvNet [VL15] package.
- **CIFAR-10:** We trained a three-layer LeNet architecture, as well as a Network In Network (NIN) architecture [LCY14].
- **ILSVRC 2012:** We used CaffeNet [Jia+14] and GoogLeNet [Sze+15] pre-trained models.

In order to evaluate the robustness to adversarial perturbations of a classifier $f$, we compute the average *normalized* robustness $\hat{\rho}_{\text{adv}}(f)$, defined by

$$\hat{\rho}_{\text{adv}}(f) = \frac{1}{|\mathscr{D}|} \sum_{\boldsymbol{x} \in \mathscr{D}} \frac{\hat{\Delta}_{\text{adv}}(\boldsymbol{x})}{\|\boldsymbol{x}\|_2}, \tag{3.15}$$

where $\hat{\Delta}_{\text{adv}}(\boldsymbol{x}) = \|\hat{\boldsymbol{r}}(\boldsymbol{x})\|_2$ is the estimated adversarial robustness at $\boldsymbol{x}$ computed using the proposed approach, and $\mathscr{D}$ denotes a set of datapoints sampled from $\mu$.[4]

To correctly assess the robustness and draw conclusions on the robustness of a classifier $f$, it is important to compare $\hat{\rho}_{\text{adv}}(f)$ to the right quantity. To do so, we define *the intrinsic "data" robustness* $\hat{\rho}_d$, which measures a distance between different classes in the dataset. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ be training datapoints, and $y(\boldsymbol{x}_1), \ldots, y(\boldsymbol{x}_m)$ denote their ground truth labels. We define the intrinsic "data" robustness to be

$$\hat{\rho}_d = \frac{1}{m} \sum_{i=1}^{m} \min_{j: y(\boldsymbol{x}_j) \neq y(\boldsymbol{x}_i)} \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2}{\|\boldsymbol{x}_i\|_2}. \tag{3.16}$$

Note that this quantity is independent of the classifier $f$, and only depends on the dataset. It represents the average normalized norm of the minimal perturbation required to "transform" a training point to a training point of another class, and can be seen as a distance measure

[3] To see this, one can apply Holder's inequality to obtain a lower bound on the $\ell_p$ norm of the perturbation.

[4] In practice, for MNIST and CIFAR-10 datasets, we set $\mathscr{D}$ to be the test data. We set $\mathscr{D}$ to be the validation set for the ILSVRC 2012 experiments.

| **Classifier** | Test error | $\hat{\rho}_{\text{adv}}$ (Ours) | time | $\hat{\rho}_{\text{adv}}$ [GSS15] | time | $\hat{\rho}_{\text{adv}}$ [Sze+14] | time | $\hat{\rho}_d$ |
|---|---|---|---|---|---|---|---|---|
| LeNet (MNIST) | 1% | $2.0 \times 10^{-1}$ | 110 ms | 1.0 | 20 ms | $2.5 \times 10^{-1}$ | $> 4$ s | $8.3 \times 10^{-1}$ |
| FC500-150-10 (MNIST) | 1.7% | $1.1 \times 10^{-1}$ | 50 ms | $3.9 \times 10^{-1}$ | 10 ms | $1.2 \times 10^{-1}$ | $> 2$ s | $8.3 \times 10^{-1}$ |
| NIN (CIFAR-10) | 11.5% | $2.3 \times 10^{-2}$ | 1100 ms | $1.2 \times 10^{-1}$ | 180 ms | $2.4 \times 10^{-2}$ | >50 s | $7.4 \times 10^{-1}$ |
| LeNet (CIFAR-10) | 22.6% | $3.0 \times 10^{-2}$ | 220 ms | $1.3 \times 10^{-1}$ | 50 ms | $3.9 \times 10^{-2}$ | >7 s | $7.4 \times 10^{-1}$ |
| CaffeNet (ILSVRC2012) | 42.6% | $2.7 \times 10^{-3}$ | 510 ms* | $3.5 \times 10^{-2}$ | 50 ms* | - | - | $4.7 \times 10^{-1}$ |
| GoogLeNet (ILSVRC2012) | 31.3% | $1.9 \times 10^{-3}$ | 800 ms* | $4.7 \times 10^{-2}$ | 80 ms* | - | - | $4.7 \times 10^{-1}$ |

Table 3.2: The adversarial robustness of different classifiers on different datasets. The time required to compute one sample for each method is given in the time columns. The times are computed on a Mid-2015 MacBook Pro without CUDA support. The asterisk marks determines the values computed using a GTX 750 Ti GPU.

between the different classes in the data set. The quantity $\hat{\rho}_d$ therefore provides a baseline for comparing the robustness to adversarial perturbations, and we say that $f$ is not robust to adversarial perturbations when $\hat{\rho}_{\text{adv}}(f) \ll \hat{\rho}_d$.

We compare the proposed approach to state-of-the-art techniques to compute adversarial perturbations, namely the methods in [Sze+14] and [GSS15]. The method in [Sze+14] solves a series of penalized optimization problems to find the minimal perturbation, whereas [GSS15] estimates the minimal perturbation by taking the sign of the gradient

$$\hat{\boldsymbol{r}}(\boldsymbol{x}) = \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right),$$

with $J$ the cost used to train the neural network, $\boldsymbol{\theta}$ is the model parameters, and $y$ is the label of $\boldsymbol{x}$. The method is called *fast gradient sign method*. In practice, in the absence of general rules to choose the parameter $\epsilon$, we chose the smallest $\epsilon$ such that 90% of the data is misclassified after perturbation.[5]

### 3.5.2 Results

We report in Table 3.2 the accuracy and average robustness $\hat{\rho}_{\text{adv}}$ of each classifier computed using different methods. We also show the running time required for each method to compute *one* adversarial sample.

It can be seen that the proposed approach estimates smaller perturbations (hence closer to the minimal perturbation defined in Eq. (3.3)) than the ones computed using the other approaches. For example, on the ILSVRC2012 challenge dataset, the average perturbation is one order of magnitude smaller compared to the fast gradient sign method. It should be noted moreover that the proposed approach also yields slightly smaller perturbation vectors than the method in [Sze+14]. The proposed approach is hence more accurate in detecting directions that can potentially fool neural networks. As a result, the proposed approach can be used as a valuable tool to accurately assess the robustness of classifiers. On the complexity aspect, the proposed algorithm is substantially faster than the standard method proposed in [Sze+14]. In fact, while the approach in [Sze+14] involves a costly minimization of a series of objective functions, we observed empirically that our algorithm converges in a

[5]Using this method, we observed empirically that one cannot reach 100% misclassification rate on some datasets. In fact, even by increasing $\epsilon$ to be very large, this method can fail in misclassifying all samples.

few iterations (i.e., less than 3) to a perturbation vector that fools the classifier. Hence, our method reaches a more accurate perturbation vector compared to state-of-the-art methods, while being computationally efficient. This makes it readily suitable to be used as a baseline method to estimate the robustness of very deep neural networks on large-scale datasets. It can be seen that, despite their very good test accuracy, these methods are extremely unstable to adversarial perturbations, as we have $\hat{\rho}_{\text{adv}}(f) \ll \hat{\rho}_d$ (where $\hat{\rho}_{\text{adv}}(f)$ is computed using the proposed method) for all tested classifiers on CIFAR-10 and ILSVRC2012 datasets. Note for example that a perturbation that is 1000 times smaller in magnitude than the original image is sufficient to fool a state-of-the-art deep neural network on the ILSVRC2012 dataset.

We illustrate in Figure 3.6 perturbed images generated by the fast gradient sign and the proposed approach. It can be observed that the proposed method generates adversarial perturbations which are hardly perceptible, while the fast gradient sign method outputs a perturbation image with higher norm.

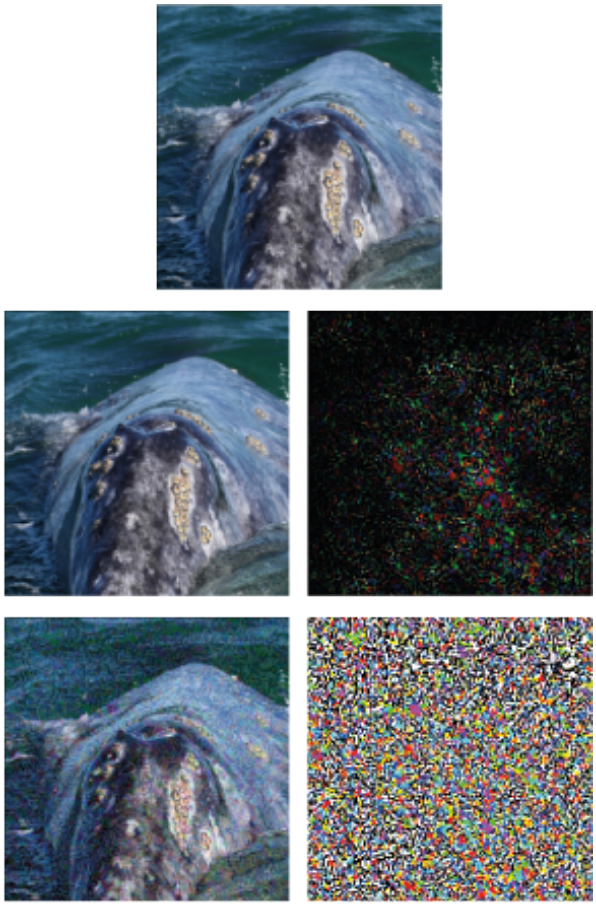


Figure 3.6: An example of adversarial perturbations. First row: the original image $\boldsymbol{x}$ that is classified as $\hat{k}(\boldsymbol{x})$="whale". Second row: the image $\boldsymbol{x} + \boldsymbol{r}$ classified as $\hat{k}(\boldsymbol{x} + \boldsymbol{r})$="turtle" and the corresponding perturbation $\boldsymbol{r}$ computed by the proposed algorithm. Third row: the image classified as "turtle" and the corresponding perturbation computed by the fast gradient sign method [GSS15]. Our approach leads to a smaller perturbation.

It should be noted that, when perturbations are measured using the $\ell_\infty$ norm, the above conclusions remain unchanged: the proposed method yields adversarial perturbations that

| **Classifier** | Proposed | Fast gradient sign |
|---|---|---|
| LeNet (MNIST) | 0.10 | 0.26 |
| FC500-150-10 (MNIST) | 0.04 | 0.11 |
| NIN (CIFAR-10) | 0.008 | 0.024 |
| LeNet (CIFAR-10) | 0.015 | 0.028 |

Table 3.3: Values of $\hat{\rho}^{\infty}_{\text{adv}}$ for four different networks based on the proposed method (smallest $\ell_\infty$ perturbation) and fast gradient sign method with 90% of misclassification.

are smaller (hence closer to the optimum) compared to other methods for computing adversarial examples. Table 3.3 reports the $\ell_\infty$ robustness to adversarial perturbations measured by $\hat{\rho}^{\infty}_{\text{adv}}(f) = \frac{1}{|\mathscr{D}|}\sum_{\boldsymbol{x}\in\mathscr{D}} \frac{\|\hat{\boldsymbol{r}}(\boldsymbol{x})\|_\infty}{\|\boldsymbol{x}\|_\infty}$, where $\hat{\boldsymbol{r}}(\boldsymbol{x})$ is computed respectively using the proposed method (with $p = \infty$, see Section 3.4.3), and the fast gradient sign method for MNIST and CIFAR-10 tasks.

**Evidence of optimality of perturbations computed using the proposed method.**

The proposed approach to solve the optimization problem in Eq. (3.3) is not exact and involves an iterative linearization of the decision boundary. The estimated perturbation is therefore only guaranteed to provide an *upper bound* on the optimal perturbation $\Delta_{\text{adv}}(\boldsymbol{x})$. To provide an accurate assessment of the classifiers' robustness, it is however important that the estimated perturbation is as close as possible to the optimal one $\Delta_{\text{adv}}(\boldsymbol{x})$, and the error due to the approximation should be small. It should further be noted that a *necessary* condition for optimality of a perturbation $\boldsymbol{r}$ in the sense of Eq. (3.3) is that $\boldsymbol{r}$ is collinear to the gradient of the boundary at $\boldsymbol{x}_0 + \boldsymbol{r}$. The following result formalizes this necessary condition:

**Fact 2.** *The optimal perturbation $\boldsymbol{r}^* := \boldsymbol{r}^*(\boldsymbol{x}_0)$ is collinear to the gradient of the decision boundary at $\boldsymbol{x}_0 + \boldsymbol{r}^*$, given by*

$$\boldsymbol{g}(\boldsymbol{x}_0 + \boldsymbol{r}^*) := \nabla(f_{\hat{k}(\boldsymbol{x}_0)} - f_{k^*})(\boldsymbol{x}_0 + \boldsymbol{r}^*),$$

*where $k^*$ denotes the closest class to $\hat{k}(\boldsymbol{x}_0)$: $k^* = \underset{k\neq\hat{k}(\boldsymbol{x}_0)}{\operatorname{argmin}} \|\boldsymbol{r}^k\|_2$, provided $g(\boldsymbol{x}_0 + \boldsymbol{r}^*) \neq \boldsymbol{0}$.*

*Proof.* Let $k \neq \hat{k}(\boldsymbol{x}_0)$. We recall that

$$\boldsymbol{r}^k = \operatorname{argmin} \|\boldsymbol{r}\|_2^2 \text{ subject to } f_k(\boldsymbol{x}_0 + \boldsymbol{r}) \geq f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0 + \boldsymbol{r}).$$

We write the gradient of the Lagrangian of above minimization problem, and we have at optimality $2\boldsymbol{r}^k + \lambda\nabla(f_{\hat{k}(\boldsymbol{x}_0)} - f_k)(\boldsymbol{x}_0 + \boldsymbol{r}^k) = 0$. Hence, $\boldsymbol{r}^k$ is collinear to the gradient of the decision boundary at $\boldsymbol{x}_0 + \boldsymbol{r}^k$. Since $\boldsymbol{r}^*$ is the vector $\boldsymbol{r}^k$ with minimal norm, we conclude the result. □

To empirically assess the optimality of the estimated perturbation $\hat{\boldsymbol{r}}(\boldsymbol{x}_0)$ obtained with Algorithm 2, we evaluate the collinearity between the gradient at the decision boundary,

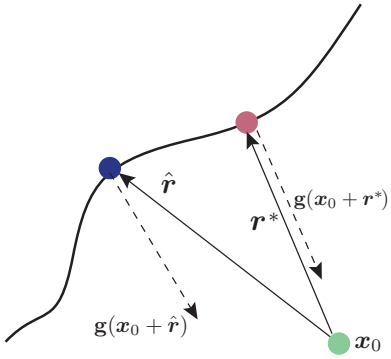



Figure 3.7: Illustration of the quantities used in the computation of Eq. (3.17). Note that for the optimal perturbation $\boldsymbol{r}^*$, we have $\mathbf{g}(\boldsymbol{x}_0 + \boldsymbol{r}^*)$ is collinear to $\boldsymbol{r}^*$. The quantity in Eq. (3.17) measures the angle between the estimated perturbation $\hat{\boldsymbol{r}}$ and the gradient vector $\mathbf{g}(\boldsymbol{x}_0 + \hat{\boldsymbol{r}})$.

$\boldsymbol{g}(\boldsymbol{x}_0 + \hat{\boldsymbol{r}}(\boldsymbol{x}_0))$ and the estimated perturbation $\hat{\boldsymbol{r}}(\boldsymbol{x}_0)$. We define the inner product $I(\boldsymbol{x}_0)$ by

$$I(\boldsymbol{x}_0) = \frac{|\langle \hat{\boldsymbol{r}}(\boldsymbol{x}_0), \boldsymbol{g}(\boldsymbol{x}_0 + \hat{\boldsymbol{r}}(\boldsymbol{x}_0))\rangle|}{\|\hat{\boldsymbol{r}}(\boldsymbol{x}_0)\|_2 \|\boldsymbol{g}(\boldsymbol{x}_0 + \hat{\boldsymbol{r}}(\boldsymbol{x}_0))\|_2}. \tag{3.17}$$

It should be noted that, when $\hat{\boldsymbol{r}}(\boldsymbol{x}_0) = \boldsymbol{r}^*(\boldsymbol{x}_0)$ is the optimal perturbation, we have $I(\boldsymbol{x}_0) = 1$, as the perturbation is collinear to the gradient to the decision boundary according to the above result. See Fig. 3.7 for an illustration. We show in Fig. 3.8 the distribution of $I(\boldsymbol{x}_0)$ obtained for random choices of images $\boldsymbol{x}_0$ from the ILSVRC2012 validation set, for the two networks CaffeNet and GoogLeNet. Interestingly, the inner product $I$ for both networks receives values close to 1. Note for example that we have $\mathbb{P}(I(\boldsymbol{x}) > 0.8) \approx 0.8$ for CaffeNet, which shows that in most examples, the estimated perturbation is approximately aligned with the gradient of the decision boundary. Note moreover that the average inner product in both cases is $\approx 0.9$. While not providing a formal proof that the perturbations obtained using our method are optimal, this experiment shows that the estimated perturbations approximately satisfy the necessary optimality condition $I(\boldsymbol{x}) = 1$.

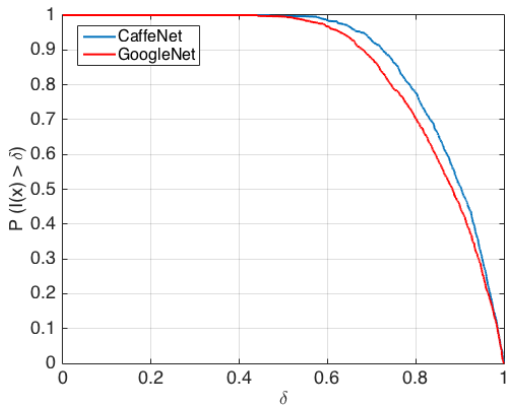



Figure 3.8: Empirical distribution of $I(\boldsymbol{x})$ (quantity defined in Eq. (3.17)) for a randomly chosen population of images $\boldsymbol{x}$ from ILSVRC 2012 validation set on CaffeNet and GoogleNet. The $y$ axis is the empirical probability that $I(\boldsymbol{x}) > \delta$, and the $\boldsymbol{x}$ axis is the threshold $\delta$.

**Fine-tuning using adversarial examples.**

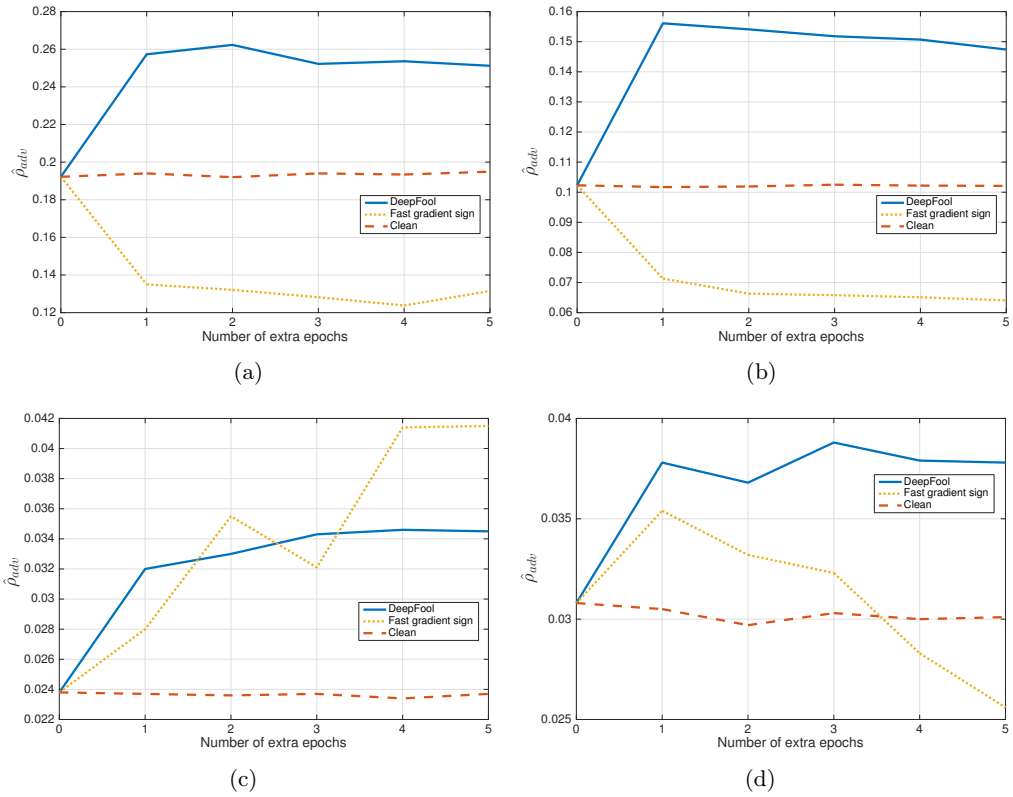



Figure 3.9: Effect of fine-tuning on adversarial examples computed by two different methods for (a) LeNet on MNIST, (b) fully-connected network on MNIST, (c) NIN on CIFAR-10, (d) LeNet on CIFAR-10. The proposed method is labeled as "DeepFool".

In this section, we fine-tune the networks of Table 3.2 on adversarial examples to build more robust classifiers for the MNIST and CIFAR-10 tasks. Specifically, for each network, we performed two experiments: (i) Fine-tuning the network on adversarial examples computed using the proposed method, (ii) Fine-tuning the network on the fast gradient sign adversarial examples. We fine-tune the networks by performing 5 additional epochs, with a 50% decreased learning rate only on the perturbed training set. For each experiment, the same training data was used through all 5 extra epochs. For the sake of completeness, we also performed 5 extra epochs on the original data. The evolution of $\hat{\rho}_{\text{adv}}$ for the different fine-tuning strategies is shown in Figures 3.9(a) to 3.9(d), *where the robustness $\hat{\rho}_{adv}$ is estimated using the proposed method*, since this is the most accurate method, as shown in Table 3.2. Observe that fine-tuning with our computed adversarial examples increases the robustness of the networks to adversarial perturbations even after one extra epoch. For example, the robustness of the networks on MNIST is improved by 50% and NIN's robustness is increased by about 40%. On the other hand, quite surprisingly, fine tuning with adversarial samples computed using the method in [GSS15] can lead to *a decreased* robustness to adversarial perturbations of the network. We hypothesize that this behavior is due to the fact that perturbations estimated using the fast gradient sign method are much larger than minimal adversarial perturbations. Fine-tuning the network with overly perturbed images decreases the robustness of the networks to adversarial perturbations. To verify this hypothesis, we compare in Figure 3.10 the adversarial robustness of a network that is fine-tuned with the adversarial examples obtained using the proposed method, where

norms of perturbations have been deliberately multiplied by $\alpha = 1, 2, 3$. Interestingly, we see that by magnifying the norms of the adversarial perturbations, the robustness of the fine-tuned network is *decreased*. This might explain why overly perturbed images decrease the robustness of MNIST networks: these perturbations can really change the actual class of the digits, hence fine-tuning based on these examples can lead to a drop of the robustness (for an illustration, see Figure 3.11). This lends credence to our hypothesis, and further shows the importance of designing accurate methods to compute minimal perturbations.

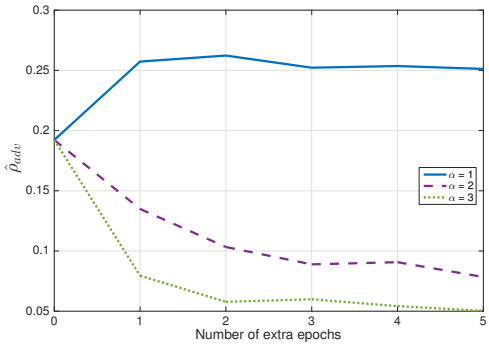



Figure 3.10: Fine-tuning based on magnified adversarial perturbations computed using our approach.


Figure 3.11: From “1" to “7" : original image classified as “1" and the perturbed images (using our approach) classified as “7" using different values of $\alpha$.

Table 3.4 lists the accuracies of the fine-tuned networks. It can be seen that fine-tuning with the proposed approach can improve the accuracy of the networks. Conversely, fine-tuning with the approach in [GSS15] has led to a decrease of the test accuracy in all our experiments. This confirms the explanation that the fast gradient sign method outputs *overly perturbed* images that lead to images that are unlikely to occur in the test data. Hence, it decreases the performance of the method as it acts as a regularizer that does not represent the distribution of the original data. This effect is analogous to the phenomena observed in geometric data augmentation schemes, where *large* transformations of the original samples have a counter-productive effect on generalization.[6]

To emphasize the importance of a correct estimation of the minimal perturbation, we now show that using approximate methods can lead to wrong conclusions regarding the adversarial robustness of networks. We fine-tune the NIN classifier on the fast gradient sign adversarial examples. We follow the procedure described earlier but this time, we decreased the learning rate by 90%. We have evaluated the adversarial robustness of this network at different extra epochs using the proposed method and the fast gradient sign method. As

[6]While the authors of [GSS15] reported an *increased* generalization performance on the MNIST task (from 0.94% to 0.84%) using adversarial regularization, it should be noted that the their experimental setup is significantly different as [GSS15] trained the network based on a modified cost function, while we performed straightforward fine-tuning solely on adversarial samples.

one can see in Figure 3.12, the red plot exaggerates the effect of training on the adversarial examples. Moreover, it is not sensitive enough to demonstrate the loss of robustness at the first extra epoch. These observations confirm that using an *accurate* tool to measure the robustness of classifiers is crucial to derive conclusions about the robustness of networks.

| **Classifier** | Proposed | Fast gradient sign | Clean |
|---|---|---|---|
| LeNet (MNIST) | 0.8% | 4.4% | 1% |
| FC500-150-10 (MNIST) | 1.5% | 4.9% | 1.7% |
| NIN (CIFAR-10) | 11.2% | 21.2% | 11.5% |
| LeNet (CIFAR-10) | 20.0% | 28.6% | 22.6% |

Table 3.4: The test error of networks after the fine-tuning on adversarial examples (after five epochs). Each columns correspond to a different type of augmented perturbation.

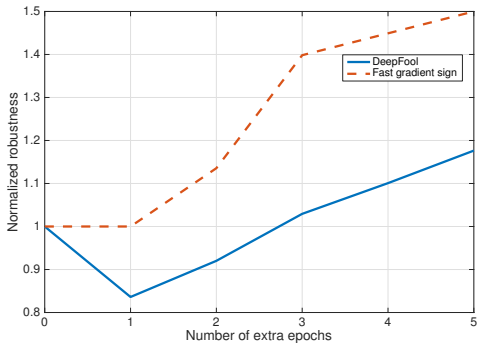



Figure 3.12: How the adversarial robustness is judged by different methods. The values are normalized by the corresponding $\hat{\rho}_{\text{adv}}$s of the original network. The proposed method is labeled as "DeepFool".

## 3.6 Conclusion

In this chapter, we proposed a new algorithm to estimate the robustness of classifiers to adversarial perturbations. It is based on an iterative linearization of the decision boundary of the classifier to generate minimal perturbations that change the estimated label of the classifier. We provided extensive experimental evidence on multiple datasets and classifiers, showing the benefits of the proposed method for computing adversarial perturbations in terms of accuracy and computational efficiency. Due to its accurate estimation of the adversarial perturbations, the proposed algorithm provides an efficient and accurate way to evaluate the robustness of classifiers and to enhance their performance by proper fine-tuning. The proposed approach can therefore be used as a building block to accurately estimate the minimal perturbation vectors, and build more robust classifiers.

Unfortunately, even with proper fine-tuning, state-of-the-art classifiers have not reached the required level of robustness, despite achieving a very good accuracy. It should be noted that several methods have recently been proposed to improve the robustness of classifiers to adversarial perturbations through regularization techniques and modified objective functions [GR14; LHL15; Hua+15], robust optimization [SYN15], foveation mechanisms [Luo+15] distillation [Pap+15] and modified activation functions [ZG16]. While these methods are

shown to yield improvements on the robustness of the deep neural networks, the design of highly robust classifiers is still an open problem. We believe that prior to achieving this goal, a fundamental question remains: does there actually exist *robust and accurate* classifiers belonging to the commonly used families of classifiers?

In the following chapter, we provide a quantitative answer to the above question, and show the existence of fundamental limits on the robustness of classifiers to perturbations, which reveals a tradeoff between robustness and risk.

# 4 Analysis of classifiers' robustness

## 4.1 Introduction

The instability of classifiers we have explored in the previous chapter raises interesting theoretical questions that we initiate in this chapter. What causes classifiers to be unstable to adversarial perturbations? Are deep networks the only classifiers that have such unstable behaviour? Is it actually feasible to learn robust *and* accurate classifiers? Providing theoretical answers to these questions is crucial in order to have a better understanding of the robustness of classifiers, and potentially to achieve better robustness. The goal of this chapter is specifically to quantify how large can the robustness to adversarial perturbations be for fixed classification families (e.g., the family of linear classifiers). To do so, we establish *learning-independent* upper bounds on the robustness of classifiers to adversarial perturbations, $\rho_{\text{adv}}(f)$, in terms of the classifier's risk $R(f)$ and data-dependent quantities. An important implication of these learning-independent limits is that in common classification cases, it is *not* possible to find a classifier in the family that achieves both a large robustness $\rho_{\text{adv}}(f)$ *and* a small risk $R(f)$, independently of the training algorithm used to choose $f$.

In more details, we assume in this chapter binary classification tasks for simplicity, and that the estimated label is hence provided by $\hat{k}_f(\boldsymbol{x}) = \text{sign}(f(\boldsymbol{x}))$. We first provide a general upper bound on the robustness of classifiers to adversarial perturbations $\rho_{\text{adv}}(f)$, and then compute specific instances of the obtained upper bound for the families of linear and quadratic classifiers. In both cases, our upper bounds are expressed in terms of the classification risk, as well as *distinguishability* measures, which depend on the considered family of classifiers, and measure informally the difficulty of the classification task with respect to the considered classifiers' family. Specifically, for linear classifiers, the distinguishability is defined as the distance between the means of the two classes, while for quadratic classifiers, it is defined as the distance between the matrices of second order moments of the two classes. For both classes of functions, our upper bounds are valid for all classifiers in the family independently of the training procedure. Our upper bounds reveal the existence of a key tradeoff between classification robustness and risk,

Part of this chapter has been published in [FFF15b]. A long version [FFF15a] is currently under review.

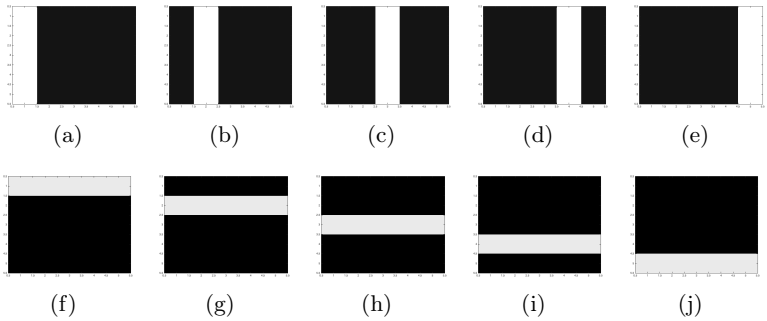



Figure 4.1: (a...e): Class 1 images. (f...j): Class -1 images.

quantified by the distinguishability quantity. This tradeoff implies in particular that in difficult classification tasks involving a small value of distinguishability, any classifier in the set with low misclassification rate is vulnerable to adversarial perturbations. Importantly, the distinguishability parameter related to quadratic classifiers is much larger than that of linear classifiers for many datasets of interest, and suggests that it is harder to find adversarial examples for more *flexible* classifiers.

The chapter is organized as follows. We introduce in Section 4.2 a simple running example used throughout the chapter to illustrate our results. Our general upper bound on the robustness to adversarial perturbations is proposed in Section 4.3, and the bound is specialized for the class of linear and quadratic classifiers in Sections 4.4 and 4.5, respectively.

## 4.2 Running example

We start our analysis with a simple running example used throughout this chapter that illustrates the notion of adversarial robustness, and highlights its difference with the notion of risk.

We consider a binary classification task on square images of size $\sqrt{d} \times \sqrt{d}$. Images of class 1 (resp. class $-1$) contain exactly one vertical line (resp. horizontal line), and a small constant positive number $a$ (resp. negative number $-a$) is added to all the pixels of the images. That is, for class 1 (resp. $-1$) images, background pixels are set to $a$ (resp. $-a$), and pixels belonging to the line are equal to $1 + a$ (resp. $1 - a$). Fig. 4.1 illustrates the classification problem for $d = 25$. The number of data points to classify is equal to $N = 2\sqrt{d}$.

Clearly, the most relevant concept (in terms of visual appearance) that permits to separate the two classes is the *orientation* of the line (i.e., horizontal vs. vertical). The *bias* of the image (i.e., the sum of all its pixels) is also a valid concept for this task, as it separates the two classes, despite being much more difficult to detect visually. The class of an image can therefore be correctly estimated from its orientation *or* from the bias. Let us first consider the linear classifier defined by

$$f_{\text{lin}}(\boldsymbol{x}) = \frac{1}{\sqrt{d}} \mathbf{1}^T \boldsymbol{x} - 1, \tag{4.1}$$

where $\mathbf{1}$ is the vector of size $d$ whose entries are all equal to 1, and $\boldsymbol{x}$ is the vectorized

image. $f_{\text{lin}}$ exploits the difference of bias between the two classes and achieves a perfect classification accuracy for all $a > 0$. Indeed, a simple computation gives $f_{\text{lin}}(\boldsymbol{x}) = \sqrt{d}a$ (resp. $f_{\text{lin}}(\boldsymbol{x}) = -\sqrt{d}a$) for class 1 (resp. class $-1$) images. Therefore, the risk of $f_{\text{lin}}$ is $R(f_{\text{lin}}) = 0$. It is important to note that $f_{\text{lin}}$ only achieves zero risk because it captures the bias, but fails to distinguish between the images based on the orientation of the line. Indeed, when $a = 0$, the data points are not linearly separable. Despite its perfect accuracy for any $a > 0$, $f_{\text{lin}}$ is *not* robust to small adversarial perturbations when $a$ is small, as a minor perturbation of the bias switches the estimated label. Indeed, a simple computation of the expected robustness leads to $\rho_{\text{adv}}(f_{\text{lin}}) = \sqrt{d}a$; therefore, the adversarial robustness of $f_{\text{lin}}$ can be made arbitrarily small by choosing $a$ to be small enough. More than that, among all linear classifiers that satisfy $R(f) = 0$, $f_{\text{lin}}$ is the one that maximizes $\rho_{\text{adv}}(f)$ (as we show later in Section 4.4). Therefore, *all* zero-risk linear classifiers are not robust to adversarial perturbations, for the classification task under consideration.

Unlike linear classifiers, a more *flexible* classifier that correctly captures the orientation of the lines in the images will be robust to adversarial perturbation, unless this perturbation significantly alters the image and modifies the direction of the line. To illustrate this point, we consider the simple setting where $d = 4$, and define the *quadratic* binary classifier

$$f_{\text{quad}}(\boldsymbol{x}) = x_1x_2 + x_3x_4 - x_1x_3 - x_2x_4, \tag{4.2}$$

where $x_1, \ldots, x_4$ denote the coordinates of $\boldsymbol{x}$ ordered from top-left pixel to bottom-right in a columnwise fashion. It should be noted that this classifier achieves zero risk (i.e., $R(f_{\text{quad}}) = 0$), as it outputs 1 for vertical line images, and $-1$ for horizontal line images. The exact computation of the adversarial robustness of quadratic classifiers is, in general, much more involved than for the linear case. However, in this simple classification example with $d = 4$, $\rho_{\text{adv}}(f_{\text{quad}})$ can be computed in closed form:

**Fact 3.** *In the above settings, the robustness to adversarial perturbations of the quadratic classifier in Eq. (4.2) satisfies* $\rho_{adv}(f_{quad}) = 1/\sqrt{2}$.

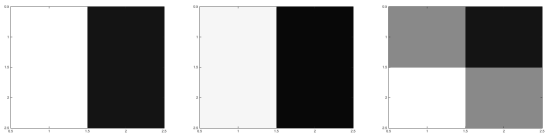
*Proof.* We have $f_{\text{quad}}(\boldsymbol{x}) = \boldsymbol{x}^T\mathbf{A}\boldsymbol{x}$, with

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix}.$$

We perform a change of basis, and work in the diagonalizing basis of $\mathbf{A}$, denoted by $\mathbf{P}$. We have

$$\mathbf{P} = \frac{1}{2}\begin{bmatrix} 1 & 1 & -1 & -1 \\ 0 & -\sqrt{2} & -\sqrt{2} & 0 \\ \sqrt{2} & 0 & 0 & \sqrt{2} \\ 1 & -1 & 1 & -1 \end{bmatrix}, \quad \mathbf{A} = \mathbf{P}^T\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}\mathbf{P}.$$

By letting $\tilde{\boldsymbol{x}} = \mathbf{P}\boldsymbol{x}$, we have $f_{\text{quad}}(\tilde{\boldsymbol{x}}) = \tilde{x}_1^2 - \tilde{x}_4^2$. Given a point $\boldsymbol{x}$ and label $y$, the following



(a) Original (b) $f_{\text{lin}}$ (c) $f_{\text{quad}}$

Figure 4.2: Robustness to adversarial noise of linear and quadratic classifiers. (a): Original image ($d = 4$, and $a = 0.1/\sqrt{d}$), (b,c): Minimally perturbed image that switches the estimated label of (b) $f_{\text{lin}}$, (c) $f_{\text{quad}}$. Note that the difference between (b) and (a) is hardly perceptible, this demonstrates that $f_{\text{lin}}$ is not robust to adversarial noise. On the other hand images (c) and (a) are clearly different, which indicates that $f_{\text{quad}}$ is more robust to adversarial noise

problem is solved to find the minimal perturbation that switches the estimated label:

$$\min_{\tilde{r}} \tilde{r}_1^2 + \tilde{r}_4^2 \text{ s.t. } y((\tilde{x}_1 + \tilde{r}_1)^2 - (\tilde{x}_4 + \tilde{r}_4)^2) \leq 0.$$

Let us consider the first datapoint $\boldsymbol{x} = [1 + a, 1 + a, a, a]^T$ (the other points can be handled in an exactly similar fashion). Then, it is easy to see that $\tilde{x}_1 = 1$ and $\tilde{x}_4 = 0$, and the minimal perturbation is achieved for $\tilde{r}_1 = -1/2$ and $\tilde{r}_4 = 1/2$. In the original space, this point corresponds to $\boldsymbol{r} = \mathbf{P}^T\tilde{\boldsymbol{r}} = [0, -1/2, 1/2, 0]^T$. Therefore, $\|\boldsymbol{r}\|_2 = 1/\sqrt{2}$, and we obtain $\rho_{\text{adv}}(f_{\text{quad}}) = 1/\sqrt{2}$. □

It should be noted that, while the robustness of the linear classifier $f_{\text{lin}}$ depends on the bias $a$, and can be very small (when $a \to 0$), the robustness of the quadratic classifier $\rho_{\text{adv}}(f_{\text{quad}})$ is much larger. This is due to the fact that the quadratic classifier exploits the most visual/strong concept that separates the two classes; that is, the orientation of the line. The robustness of both classifiers $f_{\text{lin}}$ and $f_{\text{quad}}$ is illustrated in Fig. 4.2. While a hardly perceptible change of the image is sufficient to switch the estimated label for the linear classifier, the necessary perturbation for $f_{\text{quad}}$ is a much larger one, which modifies the direction of the line to a great extent.

The above example highlights several important facts, which are summarized as follows:

- **Risk and adversarial robustness are two distinct properties of a classifier.** While $R(f_{\text{lin}}) = 0$, $f_{\text{lin}}$ is definitely not robust to small adversarial perturbations.[1] This is due to the fact that $f_{\text{lin}}$ only captures the bias in the images and ignores the orientation of the line.
- **To capture orientation (i.e., the most visual concept), one has to use a classifier that is flexible enough for the task.** Unlike the class of linear classifiers, the class of polynomial classifiers of degree 2 correctly captures the line orientation, for $d = 4$.
- **The robustness to adversarial perturbations provides a quantitative measure of the strength of a concept.** Since $\rho_{\text{adv}}(f_{\text{lin}}) \ll \rho_{\text{adv}}(f_{\text{quad}})$, one can

[1] The opposite is also possible, since a constant classifier (e.g., $f(\boldsymbol{x}) = 1$ for all $\boldsymbol{x}$) is clearly robust to perturbations, but does not achieve good accuracy.

confidently say that the concept captured by $f_{\text{quad}}$ is *stronger* than that of $f_{\text{lin}}$, in the sense that the essence of the classification task is captured by $f_{\text{quad}}$, but not by $f_{\text{lin}}$ (while they are equal in terms of misclassification rate). In general classification problems, the quantity $\rho_{\text{adv}}(f)$ provides a natural way to evaluate and compare the learned concept; larger values of $\rho_{\text{adv}}(f)$ indicate that stronger concepts are learned, for comparable values of the risk.

As illustrated in the above toy example, the robustness to adversarial perturbations is key to assess the strength of a concept. In real-world classification tasks, weak concepts correspond to partial information about the classification task (which are possibly sufficient to achieve a good accuracy), while strong concepts capture the essence of the classification task.

In the remainder of this chapter, our goal is to quantify how large can the robustness to adversarial perturbations be, for fixed classification families (e.g., family of *linear classifiers*). To do so, we establish upper bounds on the adversarial robustness $\rho_{\text{adv}}(f)$ in terms of the classifier risk $R(f)$ for all classifiers in the family. These learning-independent limits show that it is not possible to achieve a large robustness jointly with a small risk for many classification tasks of interest, independently of the training algorithm used to choose $f$.

## 4.3 Upper bound on the adversarial robustness

We now introduce our theoretical framework for analyzing the robustness to adversarial perturbations. We recall that $f : \mathbb{R}^d \to \mathbb{R}$ is a binary classifier, and that the associated decision function $\hat{k}_f(\boldsymbol{x}) = \text{sign}(f(\boldsymbol{x}))$. We recall that $\mu$ denotes the data distribution, and we denote by $\mu_1$ and $\mu_{-1}$ the probability measures of class 1 and class $-1$ datapoints, respectively. We also denote by $p_{\pm 1}$ the classwise prior probabilities, defined by $\mathop{\mathbb{P}}_{\boldsymbol{x}\sim\mu}(y(\boldsymbol{x}) = \pm 1)$, where we recall that $y(\boldsymbol{x})$ is the ground truth label of $\boldsymbol{x}$. We finally recall that the data distribution is assumed to be of bounded support; that is, $\mathop{\mathbb{P}}_{\boldsymbol{x}\sim\mu}(x \in B) = 1$, where $B = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \leq M\}$ for some $M > 0$.

We first present a key assumption on the classifier $f$ for the analysis of adversarial robustness.

**Assumption (A).** There exist $\tau > 0$ and $0 < \gamma \leq 1$ such that, for all $\boldsymbol{x} \in B$,

$$\begin{aligned} &\text{dist}(\boldsymbol{x}, S_-) \leq \tau \max(0, f(\boldsymbol{x}))^\gamma, \\ &\text{dist}(\boldsymbol{x}, S_+) \leq \tau \max(0, -f(\boldsymbol{x}))^\gamma, \end{aligned} \tag{4.3}$$

where $\text{dist}(\boldsymbol{x}, S) = \min_{\boldsymbol{z}}\{\|\boldsymbol{x} - \boldsymbol{z}\|_2 : \boldsymbol{z} \in S\}$ and $S_+$ (resp. $S_-$) is the set of points $\boldsymbol{x}$ such that $f(\boldsymbol{x}) \geq 0$ (resp. $f(\boldsymbol{x}) \leq 0$):

$$\begin{aligned} S_+ &= \{\boldsymbol{x} : f(\boldsymbol{x}) \geq 0\}, \\ S_- &= \{\boldsymbol{x} : f(\boldsymbol{x}) \leq 0\}. \end{aligned}$$

In words, the assumption (A) states that for any data point $\boldsymbol{x}$, the *residual* $\max(0, f(\boldsymbol{x}))$ (resp. $\max(0, -f(\boldsymbol{x}))$) can be used to bound the distance from $\boldsymbol{x}$ to a data point $\boldsymbol{z}$ classified

$-1$ (resp. 1).

Bounds of the form Eq. (4.3) have been established for various classes of functions since the early of work of Łojasiewicz [Łoj59] in algebraic geometry and have found applications in areas such as mathematical optimization [Pan97; LP98]. For example, Łojasiewicz [Łoj59] and later [LP94b] have shown that, remarkably, *assumption (A) holds for the general class of analytic functions*. In [NZ03], (A) is shown to hold with $\gamma = 1$ for piecewise linear functions. In [LL94], error bounds on polynomial systems are studied. Proving inequality (4.3) with explicit constants $\tau$ and $\gamma$ for different classes of functions is still an active area of research however [LMP14]. In Sections 4.4 and 4.5, we provide examples of function classes for which (A) holds, and explicit formulas for the parameters $\tau$ and $\gamma$.

The following result establishes a general upper bound on the robustness to adversarial perturbations:

**Lemma 1.** *Let $f$ be an arbitrary classifier that satisfies (A) with parameters $(\tau, \gamma)$. Then,*

$$\rho_{adv}(f) \leq 4^{1-\gamma}\tau \left(p_1 \mathbb{E}_{\mu_1}(f(\boldsymbol{x})) - p_{-1}\mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})) + 2\|f\|_\infty R(f)\right)^\gamma .$$

The proof can be found in Appendix A.1. The above result provides an upper bound on the adversarial robustness that depends on the *risk* of the classifier, as well as on a measure of the separation between the *expectations* of the classifier values computed on distribution $\mu_1$ and $\mu_{-1}$. This result is general, as we only assume that $f$ satisfies assumption $(A)$. In the next two sections, we apply Lemma 1 to two classes of classifiers, and derive interpretable upper bounds in terms of a distinguishibality measure (that depends only on the dataset) which captures the notion of difficulty of a classification task. Studying the general result in Lemma 1 through two practical classes of classifiers shows the implications of such a limit on the adversarial robustness, and illustrates the methodology for deriving family-specific upper bounds on the adversarial robustness from the above general upper bound.

## 4.4 Robustness of linear classifiers to adversarial perturbations

We define the classification function $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$. Note that any linear classifier for which $|b| > M\|\boldsymbol{w}\|_2$ is a trivial classifier that assigns the same label to all points, and we therefore assume that $|b| \leq M\|\boldsymbol{w}\|_2$.

We first show that the family of linear classifiers satisfies assumption (A), with explicit parameters $\tau$ and $\gamma$.

**Lemma 2.** *Assumption (A) holds for linear classifiers $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ with $\tau = 1/\|\boldsymbol{w}\|_2$ and $\gamma = 1$.*

*Proof.* Let $\boldsymbol{x}$ be such that $f(\boldsymbol{x}) \geq 0$. The goal is to prove that $\text{dist}(\boldsymbol{x}, S_-) \leq \tau f(\boldsymbol{x})^\gamma$ (the other inequality can be handled in a similar way). We have $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$. Thus, $\text{dist}(\boldsymbol{x}, S_-) = f(\boldsymbol{x})/\|\boldsymbol{w}\|_2 \implies \tau = 1/\|\boldsymbol{w}\|_2, \gamma = 1$. □

Using Lemma 1, we now derive an interpretable upper bound on the robustness of linear classifiers to adversarial perturbations. In particular, the following theorem bounds $\rho_{\text{adv}}(f)$ from above in terms of the first moments of the distributions $\mu_1$ and $\mu_{-1}$, and the classifier's risk:

**Theorem 1.** *Let $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ such that $|b| \leq M\|\boldsymbol{w}\|_2$. Then,*

$$\rho_{adv}(f) \leq \|p_1\mathbb{E}_{\mu_1}(\boldsymbol{x}) - p_{-1}\mathbb{E}_{\mu_{-1}}(\boldsymbol{x})\|_2 + M(|p_1 - p_{-1}| + 4R(f)). \tag{4.4}$$

*In the balanced setting where $p_1 = p_{-1} = 1/2$, and if the intercept $b = 0$ the following inequality holds:*

$$\rho_{adv}(f) \leq \frac{1}{2}\|\mathbb{E}_{\mu_1}(\boldsymbol{x}) - \mathbb{E}_{\mu_{-1}}(\boldsymbol{x})\|_2 + 2MR(f). \tag{4.5}$$

*Proof.* Using Lemma 1 with $\tau = 1/\|\boldsymbol{w}\|_2$ and $\gamma = 1$, we have

$$\rho_{\text{adv}}(f) \leq \frac{1}{\|\boldsymbol{w}\|_2}\left(\boldsymbol{w}^T\left(p_1\mathbb{E}_{\mu_1}(\boldsymbol{x}) - p_{-1}\mathbb{E}_{\mu_{-1}}(\boldsymbol{x})\right) + b(p_1 - p_{-1}) + 2\|f\|_\infty R(f)\right) \tag{4.6}$$

Observe that

i. $\boldsymbol{w}^T\left(p_1\mathbb{E}_{\mu_1}(\boldsymbol{x}) - p_{-1}\mathbb{E}_{\mu_{-1}}(\boldsymbol{x})\right) \leq \|\boldsymbol{w}\|_2\|p_1\mathbb{E}_{\mu_1}(\boldsymbol{x}) - p_{-1}\mathbb{E}_{\mu_{-1}}(\boldsymbol{x})\|_2$ using Cauchy-Schwarz inequality.
ii. $b(p_1 - p_{-1}) \leq M\|\boldsymbol{w}\|_2|p_1 - p_{-1}|$ using the assumption $|b| \leq M\|\boldsymbol{w}\|_2$,
iii. $\|f\|_\infty = \max_{\boldsymbol{x}:\|\boldsymbol{x}\|_2\leq M}\{|\boldsymbol{w}^T\boldsymbol{x} + b|\} \leq 2M\|\boldsymbol{w}\|_2$.

By plugging the three above inequalities in Eq. (4.6), we obtain the desired result in Eq. (4.4).

Finally, when $p_1 = p_{-1} = 1/2$, and the intercept $b = 0$, inequality (iii) can be tightened to $\|f\|_\infty \leq M\|\boldsymbol{w}\|_2$, and directly leads to the stated result Eq. (4.5). □

Our upper bound on $\rho_{\text{adv}}(f)$ depends on the difference of means $\|\mathbb{E}_{\mu_1}(\boldsymbol{x}) - \mathbb{E}_{\mu_{-1}}(\boldsymbol{x})\|_2$, which measures the data distinguishability between the classes. Note that this term is classifier-independent, and is only a property of the classification task. The upper bound only depends on $f$ through the risk $R(f)$. Thus, in classification tasks where the means of the two distributions are close (i.e., $\|\mathbb{E}_{\mu_1}(\boldsymbol{x}) - \mathbb{E}_{\mu_{-1}}(\boldsymbol{x})\|_2$ is small), *any linear classifier* with small risk will necessarily have a small robustness to adversarial perturbations. Note that the upper bound logically increases with the risk, as there clearly exist robust linear classifiers that achieve high risk (e.g., constant classifier). Fig. 4.3 pictorially represents the $\rho_{\text{adv}}$ vs $R$ *tradeoff diagram* as predicted by Theorem 1. Each linear classifier is represented by a point on the $\rho_{\text{adv}}$–$R$ diagram, and our result shows the existence of a region that linear classifiers cannot attain.

Quite importantly, in many interesting classification problems, the quantity $\|\mathbb{E}_{\mu_1}(\boldsymbol{x}) - \mathbb{E}_{\mu_{-1}}(\boldsymbol{x})\|_2$ is small due to large intra-class variability (e.g., due to complex intra-class geometric transformations in computer vision applications). Therefore, even if a linear

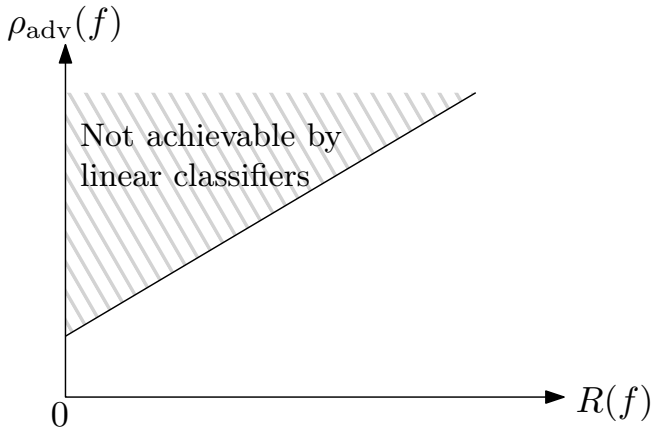



Figure 4.3: Adversarial robustness $\rho_{\text{adv}}$ versus risk diagram for linear classifiers. Each point in the plane represents a linear classifier $f$. The non-achievable zone is shown (Theorem 1). In the simplified case of Theorem 1, the intercept is equal to $\frac{1}{2}\|\mathbb{E}_{\mu_1}(\boldsymbol{x}) - \mathbb{E}_{\mu_{-1}}(\boldsymbol{x})\|_2$, and the slope is equal to $2M$.

classifier can achieve a good classification performance on such a task, it will not be robust to small adversarial perturbations. It should be noted that this limitation on the robustness to adversarial perturbations is *learning-independent*; that is, it is not possible to go beyond this fundamental limit, even if one modifies the training algorithm used to choose $f$.

#### Illustration of the results on the running example

We now illustrate our theoretical results on the example of Section 4.2. In this case, we have $\|\mathbb{E}_{\mu_1}(\boldsymbol{x}) - \mathbb{E}_{\mu_{-1}}(\boldsymbol{x})\|_2 = 2\sqrt{d}a$. By using Theorem 1, *any* zero-risk linear classifier satisfies $\rho_{\text{adv}}(f) \leq \sqrt{d}a$. As we choose $a \ll 1/\sqrt{d}$, accurate linear classifiers are therefore not robust to adversarial perturbations for this task. We note that $f_{\text{lin}}$ (defined in Eq.(4.1)) achieves the upper bound and is therefore the most robust accurate linear classifier one can get, as it can easily be checked that $\rho_{\text{adv}}(f_{\text{lin}}) = \sqrt{d}a$. In Fig. 4.4 the exact $\rho_{\text{adv}}$ vs $R$ curve is compared to our theoretical upper bound[2], for $d = 25$, $N = 10$ and a bias $a = 0.1/\sqrt{d}$. Besides the zero-risk case where our upper bound is tight, the upper bound is reasonably close to the exact curve for other values of the risk (despite not being tight).

## 4.5 Adversarial robustness of quadratic classifiers

We now study the robustness to adversarial perturbations of quadratic classifiers of the form $f(\boldsymbol{x}) = \boldsymbol{x}^T\mathbf{A}\boldsymbol{x}$, where $\mathbf{A}$ is a symmetric matrix. Besides the practical interest of quadratic classifiers in some applications [GE08; Cha+10], they represent a natural extension of linear classifiers. The study of linear vs. quadratic classifiers provides insights into how adversarial robustness depends on the family of considered classifiers. Similarly to the linear setting, we exclude the case where $f$ is a trivial classifier that assigns a constant label to all data

[2]The exact curve is computed using a bruteforce approach that enumerates all possible partitioning of the data points with linear classifiers.

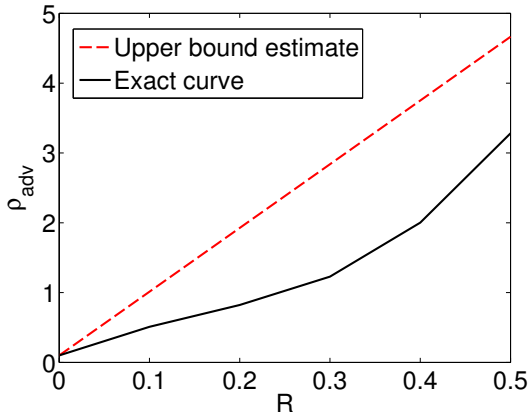



Figure 4.4: The exact $\rho_{\text{adv}}$ versus risk achievable curve, and our upper bound estimate on the running example.

points. That is, we assume that $\mathbf{A}$ satisfies

$$\lambda_{\min}(\mathbf{A}) < 0, \quad \lambda_{\max}(\mathbf{A}) > 0, \tag{4.7}$$

where $\lambda_{\min}(\mathbf{A})$ and $\lambda_{\max}(\mathbf{A})$ are the smallest and largest eigenvalues of $\mathbf{A}$. We moreover impose that the eigenvalues of $\mathbf{A}$ satisfy

$$\max\left(\left|\frac{\lambda_{\min}(\mathbf{A})}{\lambda_{\max}(\mathbf{A})}\right|, \left|\frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}\right|\right) \leq K, \tag{4.8}$$

for some constant value $K \geq 1$. This assumption imposes an approximate symmetry of the extremal eigenvalues of $\mathbf{A}$ around 0, thereby disallowing a large bias towards any of the two classes.

We first show that the assumption (A) is satisfied for quadratic classifiers, and derive explicit formulas for $\tau$ and $\gamma$.

**Lemma 3.** *Assumption (A) holds for the class of quadratic classifiers $f(\boldsymbol{x}) = \boldsymbol{x}^T\mathbf{A}\boldsymbol{x}$ where $\lambda_{\min}(\mathbf{A}) < 0$, $\lambda_{\max}(\mathbf{A}) > 0$ with $\tau = \max(|\lambda_{\min}(\mathbf{A})|^{-1/2}, |\lambda_{\max}(\mathbf{A})|^{-1/2})$, and $\gamma = 1/2$,*

*Proof.* Let $\boldsymbol{x}$ be such that $f(\boldsymbol{x}) \geq 0$, and the goal is to prove that $\text{dist}(\boldsymbol{x}, S_-) \leq \tau f(\boldsymbol{x})^\gamma$ (the other inequality can be handled in a similar way). Assume without loss of generality that $\mathbf{A}$ is diagonal (this can be done using an appropriate change of basis). Let $\nu = -\lambda_{\min}(\mathbf{A})$. We have $f(\boldsymbol{x}) = \sum_{i=1}^{d-1} \lambda_i x_i^2 - \nu x_d^2$. By setting $r_i = 0$ for all $i \in \{1, \ldots, d-1\}$ and $r_d = \text{sign}(x_d)\sqrt{f(\boldsymbol{x})/\nu}$, (where $\text{sign}(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \geq 0$ and $-1$ otherwise) we have

$$\begin{aligned} f(\boldsymbol{x}+\boldsymbol{r}) &= \sum_{i=1}^{d-1} \lambda_i x_i^2 - \nu(x_d + \text{sgn}(x_d)\sqrt{f(\boldsymbol{x})/\nu})^2 \\ &= f(\boldsymbol{x}) - 2\nu x_d \text{sgn}(x_d)\sqrt{f(\boldsymbol{x})/\nu} - f(\boldsymbol{x}) \\ &= -2\nu|x_d|\sqrt{f(\boldsymbol{x})/\nu} \leq 0. \end{aligned}$$

Hence, $\text{dist}(\boldsymbol{x}, S_-) \leq \|\boldsymbol{r}\|_2 = \nu^{-1/2}\sqrt{f(\boldsymbol{x})} \implies \tau = \nu^{-1/2}, \gamma = 1/2.$ □

The following result builds on Lemma 1 and bounds the adversarial robustness of quadratic classifiers as a function of the second order moments of the data distribution and the risk.

**Theorem 2.** *Let $f(\boldsymbol{x}) = \boldsymbol{x}^T\mathbf{A}\boldsymbol{x}$, where $\mathbf{A}$ satisfies Eqs. (4.7) and (4.8). Then,*

$$\rho_{adv}(f) \leq 2\sqrt{K\|p_1\mathbf{C}_1 - p_{-1}\mathbf{C}_{-1}\|_* + 2MKR(f)},$$

*where $\mathbf{C}_{\pm 1}(i,j) = (\mathbb{E}_{\mu_{\pm 1}}(x_i x_j))_{1\leq i,j\leq d}$, and $\|\cdot\|_*$ denotes the nuclear norm defined as the sum of the singular values of the matrix.*

*Proof.* The family of classifiers under study satisfies assumption (A) with $\tau = \max(|\lambda_{\min}(\mathbf{A})|^{-1/2}, |\lambda_{\max}(\mathbf{A})|^{-1/2})$, and $\gamma = 1/2$ (see Lemma 3). By applying Lemma 1, we have

$$\rho_{\text{adv}}(f) \leq 2\tau \left(\mathbb{E}_{\mu_1}(\boldsymbol{x}^T\mathbf{A}\boldsymbol{x}) - \mathbb{E}_{\mu_{-1}}(\boldsymbol{x}^T\mathbf{A}\boldsymbol{x}) + 2\|f\|_\infty R(f)\right)^{1/2}.$$

Observe that

i. $p_1\mathbb{E}_{\mu_1}(\boldsymbol{x}^T\mathbf{A}\boldsymbol{x}) - p_{-1}\mathbb{E}_{\mu_{-1}}(\boldsymbol{x}^T\mathbf{A}\boldsymbol{x}) = \langle \mathbf{A}, p_1\mathbf{C}_1 - p_{-1}\mathbf{C}_{-1}\rangle \leq \|\mathbf{A}\|\|p_1\mathbf{C}_1 - p_{-1}\mathbf{C}_{-1}\|_*$ using the generalized Cauchy-Schwarz inequality, where $\|\cdot\|$ and $\|\cdot\|_*$ denote respectively the spectral and nuclear matrix norms.
ii. $|f(\boldsymbol{x})| = |\boldsymbol{x}^T\mathbf{A}\boldsymbol{x}| \leq \|\mathbf{A}\|\|\boldsymbol{x}\| \leq \|\mathbf{A}\|M$,
iii. $\|\mathbf{A}\|^{1/2}\tau = \max(|\lambda_{\min}(\mathbf{A})|, |\lambda_{\max}(\mathbf{A})|)^{1/2}\max(|\lambda_{\min}(\mathbf{A})|^{-1/2}, |\lambda_{\max}(\mathbf{A})|^{-1/2}) \leq \sqrt{K}$.

Applying these three inequalities, we obtain

$$\begin{aligned}\rho_{\text{adv}}(f) &\leq 2\|\mathbf{A}\|^{1/2}\tau\left(\|p_1\mathbf{C}_1 - p_{-1}\mathbf{C}_{-1}\|_* + 2MR(f)\right)^{1/2} \\ &\leq 2\sqrt{K}\left(\|p_1\mathbf{C}_1 - p_{-1}\mathbf{C}_{-1}\|_* + 2MR(f)\right)^{1/2}.\end{aligned}$$

□

In words, the upper bound on the adversarial robustness depends on a distinguishability measure, defined by $\|\mathbf{C}_1 - \mathbf{C}_{-1}\|_*$, and the classifier's risk. In difficult classification tasks, where $\|\mathbf{C}_1 - \mathbf{C}_{-1}\|_*$ is small, any quadratic classifier with low risk that satisfies our assumptions in Eq. (4.7, 4.8) is not robust to adversarial perturbations. Similarly to the linear case, these bounds are learning-independent, and hold uniformly for any classifier in the considered family.

Note that, while the distinguishability is measured with the distance between the means of the two distributions in the linear case, it is defined here as the difference between the second order moments matrices $\|\mathbf{C}_1 - \mathbf{C}_{-1}\|_*$. Therefore, in classification tasks involving two distributions with close means, and different second order moments, any zero-risk linear classifier will not be robust to adversarial noise, while zero-risk and robust quadratic classifiers are a priori possible according to our upper bound in Theorem 2. This suggests that robustness to adversarial perturbations can be larger for more flexible classifiers, for comparable values of the risk.

| | $R(f)$ | $\rho_{\text{adv}}(f)$ | Upper bound |
|---|---|---|---|
| $f_{\text{lin}}$ | 0 | $2\ a$ [0.01] | $2\ a$ [0.01] |
| $f_{\text{quad}}$ | 0 | $1/\sqrt{2}$ [0.7] | $2\sqrt{1+4a}$ [2.02] |

Table 4.1: Summary of quantities computed for the running example of Section 4.2, with $d = 4$. In blue, we show numerical values obtained with $a = 0.005$, for easier numerical comparison.

#### Illustration of the results on the running example

We apply the obtained upper bound for the example in Section 4.2, with $d = 4$. A simple computation gives $\|\mathbf{C}_1 - \mathbf{C}_{-1}\|_* = 2 + 8a \geq 2$. Note that this term is significantly larger than the difference of means (equal to $4a$, when $d = 4$). Hence, our upper bound for quadratic classifiers is much larger than for linear classifiers; in particular, the former upper bound does *not* forbid the existence of a quadratic classifier that achieves both a large robustness to adversarial perturbations and low risk for this simple classification task. In fact, the quadratic classifier $f_{\text{quad}}$ defined in Eq.(4.2) achieves zero-risk $R(f) = 0$, and has a large robustness $\rho_{\text{adv}}$. Table 4.1 summarizes the risk, adversarial robustness and the upper bounds on the adversarial perturbations for $f_{\text{lin}}$ and $f_{\text{quad}}$. Note that for small values of $a$, we have $\rho_{\text{adv}}(f_{\text{lin}}) \ll \rho_{\text{adv}}(f_{\text{quad}})$, and the corresponding upper bound for quadratic classifiers is much larger than for the family of linear classifiers. The large difference in robustness between these two classifiers is due to the fact that $f_{\text{quad}}$ differentiates the images from their *orientation*, unlike $f_{\text{lin}}$ that uses the *bias* to distinguish them. The minimal perturbation required to switch the estimated label of $f_{\text{quad}}$ is therefore one that modifies the direction of the line, while a hardly perceptible perturbation that modifies the bias is enough to flip the label for $f_{\text{lin}}$. Note finally that while $f_{\text{lin}}$ achieves the upper bound on the robustness, $f_{\text{quad}}$ does not meet the upper bound. This result might suggest the existence of other quadratic classifiers that are more robust to adversarial perturbations, while having zero risk.

## 4.6 Experimental results

In this section, we illustrate our results on practical classification examples. Specifically, through experiments on real data, we seek to confirm the identified limit on the robustness of classifiers. We also study more general classifiers to suggest that the trends obtained with our theoretical results are not limited to linear and quadratic classifiers.

### 4.6.1 Binary classification using SVM

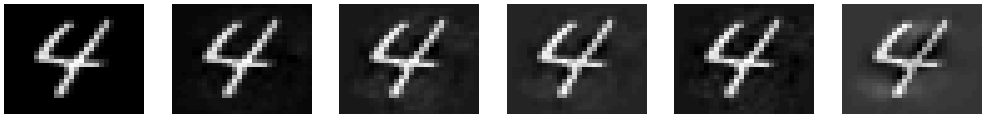
We perform experiments on several classifiers: linear SVM (denoted *L-SVM*), SVM with polynomial kernels of degree $q$ (denoted *poly-SVM (q)*), and SVM with RBF kernel with a width parameter $\sigma^2$ (*RBF-SVM($\sigma^2$)*). To train the classifiers, we use the Liblinear [Fan+08a] and LibSVM [CL11] implementations, and we fix the regularization parameters using a cross-validation procedure.

We first consider a classification task on the MNIST handwritten digits dataset [LeC+98a]. We consider a digit "4" vs. digit "5" binary classification task, with $2,000$ and $1,000$ randomly

chosen images for training and testing, respectively. In addition, a small random translation is applied to all images, and the images are normalized to be of unit Euclidean norm. Table 4.2 reports the accuracy of the different classifiers, and their robustness to adversarial perturbations estimated using the method in Chapter 3. Despite the fact that L-SVM performs fairly well on this classification task (both on training and testing), it is highly non robust to small adversarial perturbations. Indeed, $\widehat{\rho}_{\text{adv}}(f)$ is one order of magnitude smaller than the distance between the two classes $\hat{\rho}_d = 0.72$ (see Eq. (3.16) for the definition of the data robustness $\hat{\rho}_d$). Visually, this translates to an adversarial perturbation that is hardly perceptible. The instability of the linear classifier to adversarial perturbations can be predicted from Theorem 1, as the distinguishability term $\frac{1}{2}\|\mathbb{E}_{\mu_1}(\boldsymbol{x}) - \mathbb{E}_{\mu_{-1}}(\boldsymbol{x})\|_2$ is small (see Table 4.4). In addition to improving the accuracy, the more flexible classifiers are also more robust to adversarial perturbations, as predicted by our theoretical analysis. In fact, we see in Table 4.4 that the distinguishability measure for second order classifiers is much larger than for linear classifiers, hence allowing in principle for more robust classifiers in this family. It should further be noted that the third order classifier is slightly more robust than the second order one, and RBF-SVM with small width $\sigma^2 = 0.1$ is more robust than with $\sigma^2 = 1$. Note that $\sigma$ controls the flexibility of the classifier in a similar way as the degree in the polynomial kernel. Interestingly, in this relatively easy classification task, RBF-SVM(0.1) achieves both a good performance (low risk), and a high robustness to adversarial perturbations. The observations we draw from this experiment hence confirm the spirit of our theoretical analysis, where we showed that the limit on the robustness of classifiers improves with the flexibility of the family of classifiers. Fig. 4.5 illustrates the robustness of the different classifiers on an example image.

| Model | Train error (%) | Test error (%) | $\widehat{\rho}_{\text{adv}}$ |
|---|---|---|---|
| L-SVM | 4.8 | 7.0 | 0.08 |
| poly-SVM(2) | 0 | 1 | 0.19 |
| poly-SVM(3) | 0 | 0.6 | 0.24 |
| RBF-SVM(1) | 0 | 1.1 | 0.16 |
| RBF-SVM(0.1) | 0 | 0.5 | 0.32 |

Table 4.2: Training and testing accuracy of different models, and robustness to adversarial noise for the MNIST task. Note that for this example, we have $\hat{\rho}_d = 0.72$.



(a) (b) $\Delta_{\text{adv}} = 0.08$ (c) $\Delta_{\text{adv}} = 0.19$ (d) $\Delta_{\text{adv}} = 0.21$ (e) $\Delta_{\text{adv}} = 0.15$ (f) $\Delta_{\text{adv}} = 0.41$

Figure 4.5: Original image (a) and minimally perturbed images (b-f) that switch the estimated label of linear (b), quadratic (c), cubic (d), RBF(1) (e), RBF(0.1) (f) classifiers.

We now turn to a natural image classification task, with images taken from the CIFAR-10 database [KH09]. The database contains 10 classes of $32 \times 32$ RGB images. We restrict the dataset to the first two classes ("airplane" and "automobile"), and consider a subset of the original data, with $1,000$ images for training, and $1,000$ for testing. Moreover, all images are normalized to be of unit Euclidean norm. Compared to the first dataset, this task is more difficult, as the variability of the images is much larger than for digits. We report the performance and robustness results in Table 4.3. It can be seen that *all* classifiers

| Model | Train error (%) | Test error (%) | $\widehat{\rho}_{\text{adv}}$ |
|---|---|---|---|
| L-SVM | 14.5 | 21.3 | 0.04 |
| poly-SVM(2) | 4.2 | 15.3 | 0.03 |
| poly-SVM(3) | 4 | 15 | 0.04 |
| RBF-SVM(1) | 7.6 | 16 | 0.04 |
| RBF-SVM(0.1) | 0 | 13.1 | 0.06 |

Table 4.3: Training and testing accuracy of different models, and robustness to adversarial noise for the CIFAR task. Note that for this example, we have $\hat{\rho}_d = 0.39$.

| Quantity | Definition | Digits | Natural images |
|---|---|---|---|
| Distance between classes | $\hat{\rho}_d$ (see Eq. (3.16)) | 0.72 | 0.39 |
| Distinguishability (linear class.) | $\|p_1\mathbb{E}_{\mu_1}(x) - p_{-1}\mathbb{E}_{\mu_{-1}}(x)\|_2$ | 0.14 | 0.06 |
| Distinguishability (quadratic class.) | $2\sqrt{K\|p_1C_1 - p_{-1}C_{-1}\|_*}$ | 1.4 | 0.87 |

Table 4.4: The parameter $\hat{\rho}_d$, and distinguishability measures for the two classification tasks. For the numerical computation, we used $K = 1$.

are not robust to adversarial perturbations for this experiment, as $\rho_{\text{adv}}(f) \ll \hat{\rho}_d = 0.39$. Despite that, all classifiers (except L-SVM) achieve an accuracy around 85%, and a training accuracy larger than 92%. Fig. 4.6 illustrates the robustness to adversarial noise of the learned classifiers, on an example image of the dataset. Compared to the digits dataset, the distinguishability measures for this task are smaller (see Table 4.4). Our theoretical analysis therefore predicts a lower limit on the adversarial robustness of linear and quadratic classifiers for this task.

(a) (b) $\Delta_{\text{adv}} = 0.04$ (c) $\Delta_{\text{adv}} = 0.02$ (d) $\Delta_{\text{adv}} = 0.03$ (e) $\Delta_{\text{adv}} = 0.03$ (f) $\Delta_{\text{adv}} = 0.05$

Figure 4.6: Same as Fig. 4.5, but for the "airplane" vs. "automobile" classification task.

The instability of all classifiers to adversarial perturbations on this task suggests that the essence of the classification task was not correctly captured by these classifiers, even if a fairly good test accuracy is reached. To achieve better robustness, two possibilities exist: use a more flexible family of classifiers (since our theoretical results suggest that more flexible families of classifiers achieve better robustness), or use a better training algorithm for the nonlinear classifiers. The latter solution seems possible in this setting, as the theoretical limit for quadratic classifiers suggests that there is still room to improve the robustness of these classifiers.

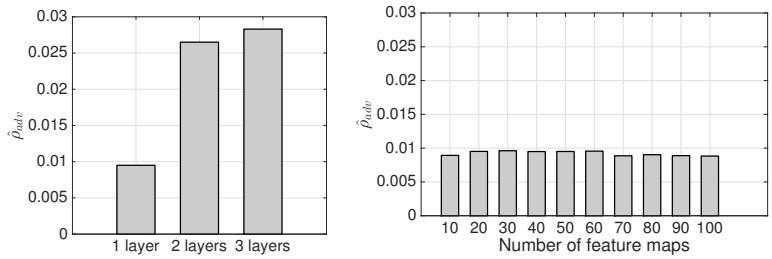
### 4.6.2 Multiclass classification using CNN

Since our theoretical results suggest that more flexible classifiers achieve better robustness to adversarial perturbations in the binary case, we now explore empirically whether the same intuitions hold for more complex classifiers (deep neural networks), and multi-class classification settings. It should be noted that, while the classifiers' flexibility is relatively well quantified for polynomial classifiers by the degree of the polynomials, this is not straightforward to do for neural network architectures. In this section, we examine the

effect of *breadth* and *depth* on the robustness to adversarial perturbations of classifiers.

We perform experiments on the multiclass CIFAR-10 classification task. We focus on baseline CNN classifiers, and learn architectures with 1, 2 and 3 hidden layers. Specifically, each layer consists of a successive combination of convolutional, rectified linear units and pooling operations. The convolutional layers consist of $5 \times 5$ filters with 50 feature maps for each layer, and the pooling operations are done on a window of size $3\times3$ with a stride parameter of 2. We build the three architectures gradually, by successively stacking a new hidden layer on top of the previous architecture (kept fixed), while re-training the entire network from scratch. The last hidden layer is then connected to a fully connected layer, and the softmax loss is used. All architectures are trained with stochastic gradient descent. To provide a fair comparison of the different classifiers, all three classifiers have approximately similar classification error (35%). To ensure similar accuracies, we perform an early stop of the training procedure when necessary. The empirical normalized robustness to adversarial perturbations $\hat{\rho}_{\text{adv}}(f)$ of the three networks are compared in Figure 4.7 (a).

We observe first that increasing the depth of the network leads to a significant increase in the robustness to adversarial perturbations, especially from 1 to 2 layers. The depth of a neural network has an important impact on the robustness of the classifier, just like the degree of a polynomial classifier is an important factor for the robustness. Going from 2 to 3 layers however seems to have a marginal effect on the robustness. It should be noted that, despite the increase of the robustness with the depth, the normalized robustness computed for all classifiers is relatively small, which suggests that none of these classifiers is really robust to adversarial perturbations. In Fig. 4.7 (b), we show the effect of the number of feature maps in the CNN (for a one layer CNN) on the estimated normalized robustness to adversarial perturbations. Unlike the effect of depth, we observe that the number of feature maps has barely any effect on the robustness to adversarial perturbations. Finally, a comparison of the normalized robustness measures of very deep networks VGG-16 and VGG-19 [SZ14] on ImageNet shows that these two networks behave very similarly in terms of robustness (both achieve a normalized robustness of $3 \cdot 10^{-3}$). This experiment, along with the experiment in Figure 4.7 (a), empirically suggest that adding layers on top of shallow network helps in terms of adversarial robustness, but if the depth of the network is already sufficiently large, then adding layers only moderately improves that robustness.




(a) Evolution with depth (b) Evolution with breadth

Figure 4.7: Evolution of the normalized robustness of classifiers with respect to (a) the depth of a CNN for CIFAR-10 task, and (b) the number of feature maps.

## 4.7 Related work & discussion

We finally discuss recent works that are related or build upon the work presented in this chapter. The topic of adversarial perturbations applied to deep networks has recently attracted a lot of attention. The authors of [GSS15] provided an empirical explanation of the phenomenon of adversarial instability. Specifically, contrarily to the original explanation provided in [Sze+14], the authors argue that it is the "linear" nature of deep nets that causes the adversarial instability. Instead, our work adopts a rigorous mathematical perspective to the problem of adversarial instability and shows that adversarial instability is due to the low flexibility of classifiers compared to the difficulty of the classification task. In [TV16], the authors provide an interesting empirical analysis of the adversarial instability, and show that adversarial examples are not isolated points, but rather occupy dense regions of the pixel space. To the best of our knowledge, our work is the first one that presents quantitative learning-independent limits on the robustness of classifiers, and shows the existence of a tradeoff between robustness and risk. The derived limits suggest that achieving a good robustness to such perturbations is more related to the *architecture* than the *learning*.

Although we did not derive a specific upper bound for feedforward deep nets with rectified activation functions due to the complexity of this architecture, we conjecture that, similarly to linear classifiers, the upper bound on the adversarial robustness is small in many tasks of interest for this family of classifiers. This is supported by empirical evidence showing that deep neural networks have properties very similar to linear classifiers (e.g., flat decision boundaries, see following chapter for extensive discussion). This would mean that the required levels of robustness to adversarial perturbations for state-of-the-art deep nets cannot be achieved with a change of the learning algorithms, and radical improvements in the robustness to adversarial perturbations would come from a change of the architecture.

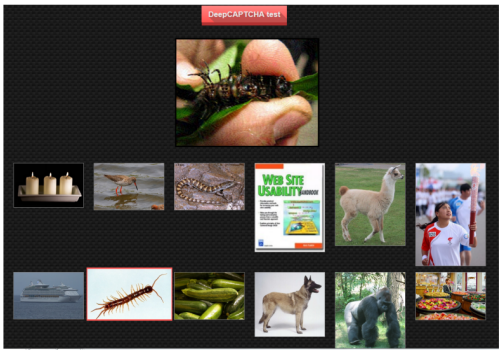



Figure 4.8: DeepCAPTCHA example. The large image is the perturbed image, and the smaller ones are the candidate images. Image taken from [Osa+16].

The existence of fundamental limits on the robustness of classifiers has recently led to an application that leverages the instability of classifiers to perturbations [Osa+16]. The authors specifically implement a CAPTCHA program, where the goal is to determine whether or not a user is human. In this work, a *perturbed* image is presented along with candidate images; the goal of the task is then to determine the image of the same class as the perturbed image in the set of candidate images (see Fig. 4.8 for an illustration). A bot that tries to solve this task would be fooled due to the perturbation presented in the original image. Note that a classifier that achieves both a large robustness and low risk

would break this CAPTCHA system, as it would correctly recognize the perturbed image, as well as the candidate images. The existence of learning-independent fundamental limits on the robustness of the family of classifiers guarantees that no classifier will have these two properties, and therefore guarantees the reliability of this CAPTCHA system.

From a security perspective, the lack of robustness of classifiers to adversarial perturbations has revealed several vulnerabilities in widely-used applications. For example, in [Car+16], the authors show that it is possible to send *hidden voice commands* to mobile phones, which are unintelligible to the human listener, but are interpreted as commands by devices. These commands, when sent from an adversarial agent, can for example order the device to make phone calls, send text messages, or go to suspicious websites, which causes significant threats to the functioning of the device.

## 4.8 Conclusions

In this chapter, we provided a quantitative analysis of the robustness of classifiers to adversarial perturbations, and showed the existence of a general upper limit on the adversarial robustness of classifiers. To better understand the implications of this limit, we derived specialized upper bounds for two families of classifiers. Our upper bounds are learning-independent, and hold for all classifiers belonging to the considered family. For the family of linear classifiers, the established limit is very small for most problems of interest. Hence, for such problems, one cannot find linear classifiers that are robust to adversarial perturbations, yet achieve a low classification risk. For the family of quadratic classifiers, the limit on adversarial robustness is usually larger than for linear classifiers, which gives hope to have classifiers that are robust to adversarial perturbations. In fact, by using an appropriate training procedure, it might be possible to get closer to the theoretical bound.

The goal of the following chapter is to study the robustness of general classifiers to more commonly encountered perturbations; that is, *random* and *semi-random* noise of the data. We precisely quantify the relation between the robustness to adversarial perturbations and these noise models and relate the obtained results to differential geometric quantities of the decision boundary.

# 5 Robustness of classifiers: from adversarial to random noise

## 5.1 Introduction

The existence of learning-independent limits on the robustness to adversarial perturbations studied in previous chapters is an important phenomenon that shows the vulnerability of classification models. These limits are however specific to *adversarial* noise, and do not apply to other noise models, such as random noise. The goal of this chapter is to study the robustness of classifiers to other noise regimes encountered in practice, and to relate these to the adversarial (or *worst-case*) perturbations studied in the previous chapter. Specifically, we precisely quantify the robustness of general nonlinear classifiers in two practical noise regimes. In the *random noise* regime, datapoints are perturbed by noise with random direction in the input space. The *semi-random* regime generalizes this model to random *subspaces* of arbitrary dimension, where an adversarial perturbation is sought within the subspace. In both cases, we derive bounds that precisely describe the robustness of classifiers as a function of the *curvature* of the decision boundary. In particular, we derive the following results:

- In the random regime, we show that the robustness of classifiers behaves as $\sqrt{d}$ times the distance from the datapoint to the classification boundary (where $d$ denotes the dimension of the data) provided the curvature of the decision boundary is sufficiently small. This result highlights the *blessing of dimensionality* for classifiers, as it implies that robustness to random noise in high dimensional classification problems *can be achieved* even for datapoints that lie very closely to the decision boundary.
- This quantification notably extends to the general semi-random regime, where we show that the robustness precisely behaves as $\sqrt{d/m}$ times the distance to decision boundary, with $m$ the dimension of the subspace. This result shows in particular that, even when $m$ is chosen as a small fraction of the dimension $d$, it is still possible to find *small* perturbations that cause data misclassification.
- We empirically show that our theoretical estimates are very accurately satisfied by state-of-the-art deep neural networks on various sets of data. This in turn leads

Part of this chapter will appear in [FMDF16].

to quantitative insights on the curvature of the decision boundary that we support experimentally through the visualization and estimation on two-dimensional sections of the boundary. In particular, the results suggest that the curvature of the decision boundary of state-of-the-art classifiers is extremely small.

This chapter is organized as follows: in Section 5.2, we define the notions of robustness to random and semi-random noise. In Section 5.3, we characterize the robustness of linear classifiers to random and semi-random noise. In Section 5.4, we show that these results also apply for general non-linear classifiers, provided the curvature of the decision boundary is kept small. Experimental results are presented in Section 5.5, where our theoretical results are shown to be very accurately satisfied by state-of-the-art deep neural networks on various sets of data. We finally conclude in Section 5.6.

## 5.2 Definitions and notations

Let $f : \mathbb{R}^d \to \mathbb{R}^L$ be an $L$-class classifier. Given a datapoint $\boldsymbol{x}_0 \in \mathbb{R}^d$, the estimated label is obtained by $\hat{k}(\boldsymbol{x}_0) = \text{argmax}_k\, f_k(\boldsymbol{x}_0)$, where $f_k(\boldsymbol{x})$ is the $k$th component of $f(\boldsymbol{x})$ that corresponds to the $k^{\text{th}}$ class. Let $\mathcal{S}$ be an arbitrary subspace of $\mathbb{R}^d$ of dimension $m$. In this chapter, we are interested in quantifying the robustness of $f$ with respect to different noise regimes. To do so, we define $\boldsymbol{r}^*_{\mathcal{S}}$ to be the perturbation in $\mathcal{S}$ of minimal norm that is required to change the estimated label of $f$ at $\boldsymbol{x}_0$;

$$\boldsymbol{r}^*_{\mathcal{S}}(\boldsymbol{x}_0) = \underset{\boldsymbol{r}\in\mathcal{S}}{\text{argmin}} \|\boldsymbol{r}\|_2 \text{ s.t. } \hat{k}(\boldsymbol{x}_0 + \boldsymbol{r}) \neq \hat{k}(\boldsymbol{x}_0), \tag{5.1}$$

and we define the robustness as

$$\Delta_{\mathcal{S}}(\boldsymbol{x}_0) := \|\boldsymbol{r}^*_{\mathcal{S}}(\boldsymbol{x}_0)\|_2.$$

Note that $\boldsymbol{r}^*_{\mathcal{S}}(\boldsymbol{x}_0)$ can be equivalently written

$$\boldsymbol{r}^*_{\mathcal{S}}(\boldsymbol{x}_0) = \underset{\boldsymbol{r}\in\mathcal{S}}{\text{argmin}} \|\boldsymbol{r}\|_2 \text{ s.t. } \exists k \neq \hat{k}(\boldsymbol{x}_0) : f_k(\boldsymbol{x}_0 + \boldsymbol{r}) \geq f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0 + \boldsymbol{r}). \tag{5.2}$$

When $\mathcal{S} = \mathbb{R}^d, \boldsymbol{r}^*(\boldsymbol{x}_0) := \boldsymbol{r}^*_{\mathbb{R}^d}(\boldsymbol{x}_0)$ is the *adversarial (or worst-case) perturbation* previously defined in Eq.(3.2), which corresponds to the (unconstrained) perturbation of minimal norm that changes the label of the datapoint $\boldsymbol{x}_0$. In other words, $\Delta_{\text{adv}}(\boldsymbol{x}_0) = \|\boldsymbol{r}^*(\boldsymbol{x}_0)\|_2$ corresponds to the minimal distance from $\boldsymbol{x}_0$ to the classifier boundary. In the case where $\mathcal{S} \subset \mathbb{R}^d$, only perturbations along $\mathcal{S}$ are allowed. The robustness of $f$ at $\boldsymbol{x}_0$ along $\mathcal{S}$ is naturally measured by the norm $\Delta_{\mathcal{S}}(\boldsymbol{x}_0) = \|\boldsymbol{r}^*_{\mathcal{S}}(\boldsymbol{x}_0)\|_2$. Different choices for $\mathcal{S}$ allow us to study the robustness of $f$ in two different regimes:

- **Random noise regime**: This corresponds to the case where $\mathcal{S}$ is a *one-dimensional subspace* ($m = 1$) with direction $\boldsymbol{v}$, where $\boldsymbol{v}$ is a *random vector* sampled uniformly from the unit sphere $\mathbb{S}^{d-1}$. Writing it explicitly, we study in this regime the robustness quantity defined by $\min_t |t|$ s.t. $\exists k \neq \hat{k}(\boldsymbol{x}_0), f_k(\boldsymbol{x}_0 + t\boldsymbol{v}) \geq f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0 + t\boldsymbol{v})$, where $\boldsymbol{v}$ is a vector sampled uniformly at random from the unit sphere $\mathbb{S}^{d-1}$.
- **Semi-random noise regime**: In this case, the subspace $\mathcal{S}$ is chosen *randomly*, but

can be of arbitrary dimension $m$.[1] We use the *semi*-random terminology as the subspace is chosen randomly, and the smallest vector that causes misclassification is then sought in the subspace. It should be noted that the random noise regime is a special case of the semi-random regime with a subspace of dimension $m = 1$. We differentiate nevertheless between these two regimes for clarity, as the perturbation direction is chosen in a worst-case fashion for subspaces of dimension larger than 1, whereas the random noise regime does not involve any choice in the direction.

In the remainder of this chapter, the goal is to establish relations between the robustness in the random and semi-random regimes on the one hand, and the robustness to adversarial perturbations $\Delta_{\text{adv}}$ on the other hand. We recall that the latter quantity captures the distance from $\boldsymbol{x}_0$ to the classifier boundary, and is therefore a key quantity in the analysis of robustness.

In the following analysis, we *fix* $\boldsymbol{x}_0$ to be a datapoint classified as $\hat{k}(\boldsymbol{x}_0)$. To simplify the notation, we remove the explicit dependence on $\boldsymbol{x}_0$ in our notations (e.g., we use $\Delta_{\mathcal{S}}$ instead of $\Delta_{\mathcal{S}}(\boldsymbol{x}_0)$ and $\hat{k}$ instead of $\hat{k}(\boldsymbol{x}_0)$), and it should be implicitly understood that all our quantities pertain to the fixed datapoint $\boldsymbol{x}_0$.

## 5.3 Robustness of affine classifiers

We first assume that $f$ is an affine classifier, i.e., $f(\boldsymbol{x}) = \mathbf{W}^\top \boldsymbol{x} + \boldsymbol{b}$ for a given $\mathbf{W} = [\boldsymbol{w}_1 \dots \boldsymbol{w}_L]$ and $\boldsymbol{b} \in \mathbb{R}^L$.

The following result shows a precise relation between the robustness to semi-random noise, $\Delta_{\mathcal{S}}$ and the robustness to adversarial perturbations, $\Delta_{\text{adv}}$.

**Theorem 3.** *Let $\mathcal{S}$ be a random $m$-dimensional subspace of $\mathbb{R}^d$, and $f$ be a $L$-class affine classifier. Let*

$$\zeta_1(m,\delta) = \left(1 + 2\sqrt{\frac{\ln(1/\delta)}{m}} + \frac{2\ln(1/\delta)}{m}\right)^{-1}, \tag{5.3}$$

$$\zeta_2(m,\delta) = \left(\max\left((1/e)\delta^{2/m}, 1 - \sqrt{2(1-\delta^{2/m})}\right)\right)^{-1}. \tag{5.4}$$

*The following inequalities hold between the robustness to semi-random noise $\Delta_{\mathcal{S}}$, and the robustness to adversarial perturbations $\Delta_{adv}$:*

$$\sqrt{\zeta_1(m,\delta)}\sqrt{\frac{d}{m}}\Delta_{adv} \leq \Delta_{\mathcal{S}} \leq \sqrt{\zeta_2(m,\delta)}\sqrt{\frac{d}{m}}\Delta_{adv}, \tag{5.5}$$

*with probability exceeding $1 - 2(L+1)\delta$.*

The proof can be found in Appendix B. Our upper and lower bounds depend on the functions $\zeta_1(m,\delta)$ and $\zeta_2(m,\delta)$ that control the inequality constants (for $m$, $\delta$ fixed). It should be

[1] A random subspace is defined as the span of $m$ independent vectors drawn uniformly at random from $\mathbb{S}^{d-1}$.

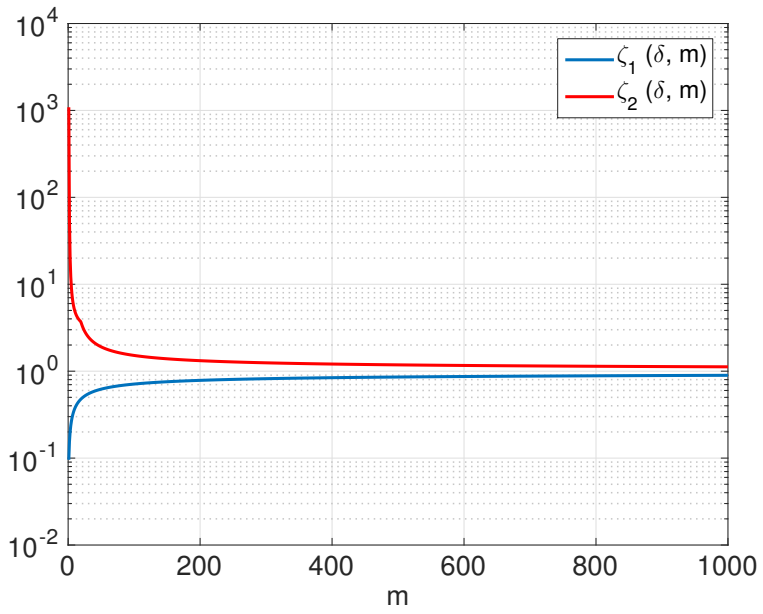



Figure 5.1: $\zeta_1(m, \delta)$ and $\zeta_2(m, \delta)$ with $\delta = 0.05$ in function of $m$.

noted that $\zeta_1(m, \delta)$ and $\zeta_2(m, \delta)$ are independent of the data dimension $d$. Figure 5.1 shows the plots of $\zeta_1(m, \delta)$ and $\zeta_2(m, \delta)$ as functions of $m$, for a fixed $\delta$. For sufficiently large $m$, $\zeta_1(m, \delta)$ and $\zeta_2(m, \delta)$ are very close to 1 (e.g., $\zeta_1(m, \delta)$ and $\zeta_2(m, \delta)$ belong to the interval $[0.8, 1.3]$ for $m \geq 250$ in the settings of Figure 5.1). The interval $[\zeta_1(m, \delta), \zeta_2(m, \delta)]$ is however (unavoidably) larger when $m = 1$.

The result in Theorem 3 shows that in the random and semi-random noise regimes, the robustness to noise is precisely related to $\Delta_{\text{adv}}$ by a $\sqrt{d/m}$ factor. Specifically, in the random noise regime ($m = 1$), the magnitude of the noise required to misclassify the datapoint behaves as $\Theta(\sqrt{d}\Delta_{\text{adv}})$ with high probability, with constants in the interval $[\zeta_1(1, \delta), \zeta_2(1, \delta)]$. Our results therefore show that, in high dimensional classification settings, affine classifiers can be robust to random noise, even if the datapoint lies very closely to the decision boundary (i.e., $\Delta_{\text{adv}}$ is small). In the semi-random noise regime with $m$ sufficiently large (e.g., $m \geq 250$), we have $\Delta_{\mathcal{S}} \approx \sqrt{d/m}\Delta_{\text{adv}}$ with high probability, as the constants $\zeta_1(m, \delta) \approx \zeta_2(m, \delta) \approx 1$ for sufficiently large $m$. Our bounds therefore "interpolate" between the random noise regime, which behaves as $\sqrt{d}\Delta_{\text{adv}}$, and the worst-case noise $\Delta_{\text{adv}}$. More importantly, the square root dependence is also notable here, as it shows that the semi-random robustness remains small even in regimes where $m$ is chosen to be a very small fraction of $d$. For example, choosing a small subspace of dimension $m = 0.01d$ results in semi-random robustness of $10\Delta_{\text{adv}}$ with high probability, which might still not be perceptible in complex visual tasks. Hence, for semi-random noise that is mostly random and only mildly adversarial, affine classifiers remain vulnerable to such noise.

## 5.4 Robustness of general classifiers

### 5.4.1 Decision boundary curvature

We now consider the general case where $f$ is a nonlinear classifier. We derive relations between the random and semi-random robustness $\Delta_{\mathcal{S}}$ and worst-case robustness $\Delta_{\text{adv}}$ using properties of the classifier's *boundary*. Let $i$ and $j$ be two arbitrary classes; we define the pairwise boundary $\mathscr{B}_{i,j}$ as the boundary of the *binary* classifier where only classes $i$ and $j$ are considered. Formally, the decision boundary $\mathscr{B}_{i,j}$ reads as follows:

$$\mathscr{B}_{i,j} = \{\boldsymbol{x} \in \mathbb{R}^d : f_i(\boldsymbol{x}) - f_j(\boldsymbol{x}) = 0\}.$$

The boundary $\mathscr{B}_{i,j}$ separates between two regions of $\mathbb{R}^d$, namely $\mathcal{R}_i$ and $\mathcal{R}_j$, where the estimated label of the binary classifier is respectively $i$ and $j$. Specifically, we have

$$\mathcal{R}_i = \{\boldsymbol{x} \in \mathbb{R}^d : f_i(\boldsymbol{x}) > f_j(\boldsymbol{x})\},$$
$$\mathcal{R}_j = \{\boldsymbol{x} \in \mathbb{R}^d : f_j(\boldsymbol{x}) > f_i(\boldsymbol{x})\}.$$

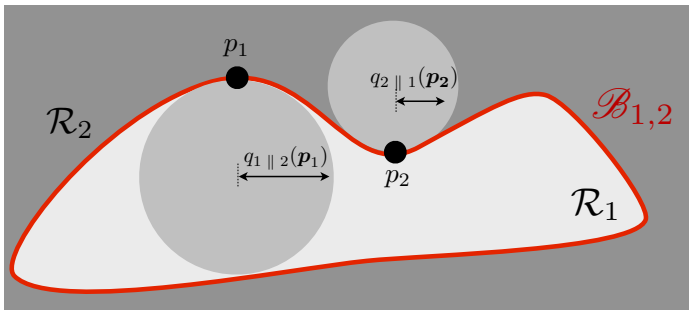



Figure 5.2: Illustration of the quantities introduced for the definition of the curvature of the decision boundary.

We assume for the purpose of this analysis that the boundary $\mathscr{B}_{i,j}$ is smooth. We now define the *curvature* of the decision boundary, denoted $\kappa(\mathscr{B}_{i,j})$, which measures how $\mathscr{B}_{i,j}$ bends in different directions in the space $\mathbb{R}^d$. For a given $\boldsymbol{p} \in \mathscr{B}_{i,j}$, we define $q_{i \parallel j}(\boldsymbol{p})$ to be the radius of the largest open ball included in the region $\mathcal{R}_i$ that intersects with $\mathscr{B}_{i,j}$ at $\boldsymbol{p}$; i.e.,

$$q_{i \parallel j}(\boldsymbol{p}) = \sup_{\boldsymbol{z} \in \mathbb{R}^d} \{\|\boldsymbol{z} - \boldsymbol{p}\|_2 : \mathbb{B}(\boldsymbol{z}, \|\boldsymbol{z} - \boldsymbol{p}\|_2) \subseteq \mathcal{R}_i\}, \tag{5.6}$$

where $\mathbb{B}(\boldsymbol{z}, \|\boldsymbol{z} - \boldsymbol{p}\|_2)$ is the open ball in $\mathbb{R}^d$ of center $\boldsymbol{z}$ and radius $\|\boldsymbol{z} - \boldsymbol{p}\|_2$. An illustration of this quantity in two dimensions is provided in Fig. 5.2. It is not hard to see that any ball $\mathbb{B}(\boldsymbol{z}^*, \|\boldsymbol{z}^* - \boldsymbol{p}\|_2)$ centered in $\boldsymbol{z}^*$ and included in $\mathcal{R}_i$ will have its tangent space at $\boldsymbol{p}$ *coincide* with the tangent of the decision boundary at the same point. This is shown as follows:

**Fact 4.** *Let $\boldsymbol{p} \in \mathscr{B}_{i,j}$, and $\boldsymbol{z}^*$ be such that $\mathbb{B}(\boldsymbol{z}^*, \|\boldsymbol{z}^* - \boldsymbol{p}\|_2) \subseteq \mathcal{R}_i$. Assuming that $\boldsymbol{p}$ is a non-singular point (i.e., $\nabla(f_i - f_j)(\boldsymbol{p}) \neq \boldsymbol{0}$), we have $\boldsymbol{z}^* - \boldsymbol{p}$ is collinear to the gradient of the decision boundary at $\boldsymbol{p}$, $\nabla(f_i - f_j)(\boldsymbol{p})$, and normal to $T_{\boldsymbol{p}}(\mathscr{B}_{i,j})$, the tangent space to the decision boundary at $\boldsymbol{p}$.*

*Proof.* We first show that $\boldsymbol{p}$ is a closest point to $\boldsymbol{z}^*$ on the decision boundary. In fact, suppose by contradiction that there exists $\boldsymbol{p}'$ such that $\|\boldsymbol{z}^* - \boldsymbol{p}'\|_2 < \|\boldsymbol{z}^* - \boldsymbol{p}\|_2$ where $\boldsymbol{p}'$ is on the boundary. Then, clearly $\boldsymbol{p}'$ belongs to the ball $\mathbb{B}(\boldsymbol{z}^*, \|\boldsymbol{z}^* - \boldsymbol{p}\|_2)$, which is by assumption included in $\mathcal{R}_i$. This raises a contradiction, as $\boldsymbol{p}' \notin \mathcal{R}_i$. Thus, we have the following relation:

$$\boldsymbol{p} - \boldsymbol{z}^* \in \operatorname*{argmin}_{\boldsymbol{r}} \|\boldsymbol{r}\|_2^2 \text{ subject to } (f_i - f_j)(\boldsymbol{z}^* + \boldsymbol{r}) = 0.$$

By writing the Lagrangian of the above minimization problem, we have $\mathcal{L}(\boldsymbol{r}, \lambda) = 2\boldsymbol{r} + \lambda \nabla (f_i - f_j)(\boldsymbol{z}^* + \boldsymbol{r})$. Equating this equation to 0 at $\boldsymbol{p} - \boldsymbol{z}^*$, we obtain that $\boldsymbol{p} - \boldsymbol{z}^*$ is collinear to $\nabla (f_i - f_j)(\boldsymbol{p})$. The latter vector is moreover orthogonal to the tangent space to the decision boundary at $\boldsymbol{p}$ by definition. □

It should further be noted that the definition in Eq. (5.6) is not symmetric in $i$ and $j$; i.e., $q_{i \parallel j}(\boldsymbol{p}) \neq q_{j \parallel i}(\boldsymbol{p})$ as the radius of the largest ball one can inscribe in both regions need not be equal. We therefore define the following symmetric quantity $q_{i,j}(\boldsymbol{p})$, where the worst-case ball inscribed in any of the two regions is considered:

$$q_{i,j}(\boldsymbol{p}) = \min \left( q_{i \parallel j}(\boldsymbol{p}), q_{j \parallel i}(\boldsymbol{p}) \right).$$

This definition describes the curvature of the decision boundary locally at $\boldsymbol{p}$ by fitting the largest ball included in one of the regions. To measure the global curvature, the worst-case radius is taken over all points on the decision boundary, i.e.,

$$q(\mathscr{B}_{i,j}) = \inf_{\boldsymbol{p} \in \mathscr{B}_{i,j}} q_{i,j}(\boldsymbol{p}), \tag{5.7}$$

$$\kappa(\mathscr{B}_{i,j}) = \frac{1}{q(\mathscr{B}_{i,j})}. \tag{5.8}$$

The curvature $\kappa(\mathscr{B}_{i,j})$ is simply defined as the inverse of the worst-case radius over all points $\boldsymbol{p}$ on the decision boundary.

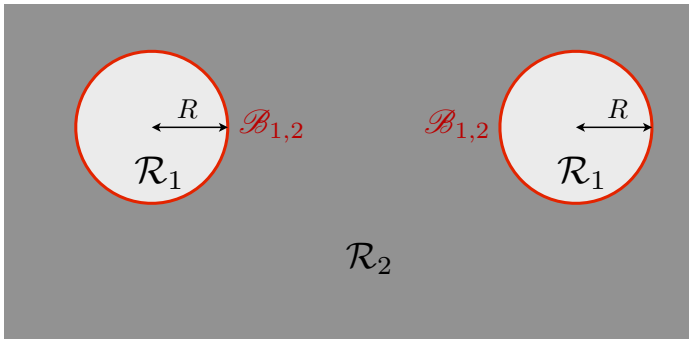



Figure 5.3: Binary classification example where the boundary is a union of two sufficiently distant spheres. In this case, the curvature is $\kappa(\mathscr{B}_{i,j}) = 1/R$, where $R$ is the radius of the circles.

In the case of affine classifiers, we have $\kappa(\mathscr{B}_{i,j}) = 0$, as it is possible to inscribe balls of infinite radius inside each region of the space. When the classification boundary is a union of (sufficiently distant) spheres with equal radius $R$ (see Fig. 5.3), the curvature

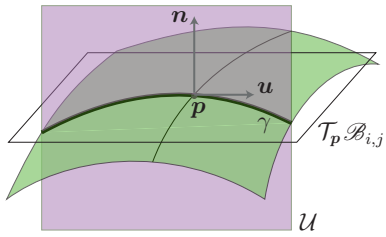



Figure 5.4: Normal section of the boundary $\mathscr{B}_{i,j}$ with respect to plane $\mathcal{U} = \text{span}(\boldsymbol{n}, \boldsymbol{u})$, where $\boldsymbol{n}$ is the normal to the boundary at $\boldsymbol{p}$, and $\boldsymbol{u}$ is an arbitrary in the tangent space $\mathcal{T}_{\boldsymbol{p}}(\mathscr{B}_{i,j})$.

$\kappa(\mathscr{B}_{i,j}) = \frac{1}{R}$. In general, the quantity $\kappa(\mathscr{B}_{i,j})$ provides an intuitive way of describing the nonlinearity of the decision boundary by fitting balls inside the classification regions.

It should be noted that there are many existing notions of curvature that one can define on hypersurfaces [Lee09]. In the simple case of a curve in a two-dimensional space, the curvature is defined as the inverse of the radius of the so-called oscullating circle. One way to define curvature for high-dimensional hypersurfaces is by taking *normal* sections of the hypersurface, and looking at the curvature of the resulting planar curve (see Fig. 5.4). Our notion of curvature is similar in spirit, but captures the *global* bending of the decision boundary by inscribing balls in the regions separated by the decision boundary.

In the following section, we show a precise characterization of the robustness to semi-random and random noise of nonlinear classifiers in terms of the curvature of the decision boundaries $\kappa(\mathscr{B}_{i,j})$.

### 5.4.2 Robustness to random and semi-random noise

We now establish bounds on the robustness to random and semi-random noise in the binary classification case. Let $\boldsymbol{x}_0$ be a datapoint classified as $\hat{k} = \hat{k}(\boldsymbol{x}_0)$. We first study the binary classification problem, where only classes $\hat{k}$ and $k \in \{1, \dots, L\} \backslash \{\hat{k}\}$ are considered. To simplify the notation, we let $\mathscr{B}_k := \mathscr{B}_{k,\hat{k}}$ be the decision boundary between classes $k$ and $\hat{k}$. In the case of the binary classification problem where classes $k$ and $\hat{k}$ are considered, the semi-random robustness and adversarial (or worst-case) robustness defined in Eq. (5.2) are given as follows:

$$\begin{aligned} \boldsymbol{r}^k_{\mathcal{S}} &= \underset{\boldsymbol{r} \in \mathcal{S}}{\text{argmin}} \|\boldsymbol{r}\|_2 \text{ s.t. } f_k(\boldsymbol{x}_0 + \boldsymbol{r}) \geq f_{\hat{k}}(\boldsymbol{x}_0 + \boldsymbol{r}), \\ \boldsymbol{r}^k &= \underset{\boldsymbol{r}}{\text{argmin}} \|\boldsymbol{r}\|_2 \text{ s.t. } f_k(\boldsymbol{x}_0 + \boldsymbol{r}) \geq f_{\hat{k}}(\boldsymbol{x}_0 + \boldsymbol{r}). \end{aligned} \tag{5.9}$$

For a randomly chosen subspace, $\Delta^k_{\mathcal{S}} := \|\boldsymbol{r}^k_{\mathcal{S}}\|_2$ is the random or semi-random robustness of the classifier, in the setting where only the two classes $k$ and $\hat{k}$ are considered. Likewise, $\Delta^k_{\text{adv}} := \|\boldsymbol{r}^k\|_2$ denotes the worst-case robustness in this setting. It should be noted that the global quantities $\boldsymbol{r}^*_{\mathcal{S}}$ and $\boldsymbol{r}^*$ are obtained from $\boldsymbol{r}^k_{\mathcal{S}}$ and $\boldsymbol{r}^k$ by taking the vectors with minimum norm over all classes $k$.

The following result gives upper and lower bounds on the ratio $\frac{\Delta^k_{\mathcal{S}}}{\Delta^k_{\text{adv}}}$ in function of the

curvature of the boundary separating class $k$ and $\hat{k}$.

**Theorem 4.** *Let $\mathcal{S}$ be a random $m$-dimensional subspace of $\mathbb{R}^d$. Let $\kappa := \kappa(\mathscr{B}_k)$. Assuming that the curvature satisfies*

$$\kappa \leq \frac{C}{\zeta_2(m,\delta)\Delta^k_{adv}}\frac{m}{d},$$

*the following inequality holds between the semi-random robustness $\Delta^k_{\mathcal{S}}$ and the adversarial robustness $\Delta^k_{adv}$:*

$$\left(1 - C_1\|\boldsymbol{r}^k\|_2\kappa\zeta_2(m,\delta)\frac{d}{m}\right)\sqrt{\zeta_1(m,\delta)}\sqrt{\frac{d}{m}} \leq \frac{\Delta^k_{\mathcal{S}}}{\Delta^k_{adv}} \leq \left(1 + C_2\|\boldsymbol{r}^k\|_2\kappa\zeta_2(m,\delta)\frac{d}{m}\right)\sqrt{\zeta_2(m,\delta)}\sqrt{\frac{d}{m}} \tag{5.10}$$

*with probability larger than $1 - 4\delta$. We recall that $\zeta_1(m,\delta)$ and $\zeta_2(m,\delta)$ are defined in Eq. (5.3, 5.4). The constants can be taken $C = 0.2, C_1 = 0.625, C_2 = 2.25$.*

The proof can be found in Appendix B. This result shows that the bounds relating the robustness to random and semi-random noise to the worst-case robustness can be extended to nonlinear classifiers, provided the curvature of the boundary $\kappa(\mathscr{B}_k)$ is sufficiently small. In the case of linear classifiers, we have $\kappa(\mathscr{B}_k) = 0$, and we recover the result for affine classifiers in Theorem 3.

To extend this result to multi-class classification, special care has to be taken. In particular, if $k$ denotes a class that has no boundary with class $\hat{k}$, we have $\Delta^k_{\text{adv}} = \infty$, and the previous curvature condition cannot be satisfied. It is therefore crucial to *exclude* such classes that have no boundary, or more generally, boundaries that are far from class $\hat{k}$. We define the set $A$ of excluded classes $k$ where $\Delta^k_{\text{adv}}$ is large

$$A = \{k : \Delta^k_{\text{adv}} \geq 1.45\sqrt{\zeta_2(m,\delta)}\sqrt{\frac{d}{m}}\|\boldsymbol{r}^*\|_2\}. \tag{5.11}$$

Note that $A$ is independent of $\mathcal{S}$, and depends only on $d$, $m$ and $\delta$. Moreover, the constants in (5.11) were chosen for simplicity of exposition.

Assuming a curvature constraint *only on the close enough classes*, the following result establishes a simplified relation between $\Delta_{\mathcal{S}}$ and $\Delta_{\text{adv}}$.

**Corollary 1.** *Let $\mathcal{S}$ be a random $m$-dimensional subspace of $\mathbb{R}^d$. Assume that, for all $k \notin A$, we have*

$$\kappa(\mathscr{B}_k)\Delta^k_{adv} \leq \frac{0.2}{\zeta_2(m,\delta)}\frac{m}{d} \tag{5.12}$$

*Then, we have*

$$0.875\sqrt{\zeta_1(m,\delta)}\sqrt{\frac{d}{m}}\Delta_{adv} \leq \Delta_{\mathcal{S}} \leq 1.45\sqrt{\zeta_2(m,\delta)}\sqrt{\frac{d}{m}}\Delta_{adv} \tag{5.13}$$

*with probability larger than $1 - 4(L+2)\delta$.*

Under the curvature condition in (5.12) on the boundaries between $\hat{k}$ and classes in $A^c$, our result shows that the robustness to random and semi-random noise exhibits the same behavior that has been observed earlier for linear classifiers in Theorem 3. In particular, $\Delta_{\mathcal{S}}$ is precisely related to the adversarial robustness $\Delta_{\text{adv}}$ by a factor of $\sqrt{d/m}$. In the random regime ($m = 1$), this factor becomes $\sqrt{d}$, and shows that in high dimensional classification problems, classifiers with sufficiently flat boundaries are robust to random noise, even if the image lies very closely to the decision boundary (i.e., $\Delta_{\text{adv}}$ is small). However, in the semi-random regime where an adversarial perturbation is found on a randomly chosen subspace of dimension $m$, the $\sqrt{d/m}$ factor that relates $\Delta_{\mathcal{S}}$ to $\Delta_{\text{adv}}$ shows that robustness to semi-random noise might not be achieved even if $m$ is chosen to be a tiny fraction of $d$ (e.g., $m = 0.01d$). In other words, if a classifier is highly vulnerable to adversarial perturbations, then it is also vulnerable to noise that is overwhelmingly random and only mildly adversarial.

It is important to note that the curvature condition in (5.12) is *not* an assumption on the curvature of the global decision boundary, but rather an assumption on the decision boundaries between pairs of classes. The distinction here is significant, as junction points where two decision boundaries meet might actually have a very large (or infinite) curvature (even in linear classification settings), and the curvature condition in (5.12) typically does not hold for this global curvature definition.

We finally stress that our results in Theorem 4 and Corollary 1 are applicable to *any* classifier, provided the decision boundaries are smooth. If we assume further prior knowledge on the considered family of classifiers and their decision boundaries (e.g., the decision boundary is a union of spheres in $\mathbb{R}^d$), similar bounds can further be derived under less restrictive curvature conditions (compared to Eq. (5.12)).

## 5.5 Experiments

### 5.5.1 Estimation of the semi-random robustness

Before delving into the experimental results to confirm our analysis, we first show that the framework developed in Chapter 3 to estimate the robustness to adversarial perturbations $\Delta_{\text{adv}}(\boldsymbol{x}_0) = \|\boldsymbol{r}^*(\boldsymbol{x}_0)\|_2$ can be modified in a straightforward way to estimate the robustness in the semi-random noise regime. We recall that Algorithm 2 operates by iteratively linearizing the classifier's decision function; the update steps derived for the affine classifiers (given in closed form) are then used at each iteration of the algorithm. Hence, to extend Algorithm 2 in the case where the perturbation is constrained to a subspace $\mathcal{S}$, we first derive the closed form formulas for the optimal subspace perturbations in the case where the classifier is affine.

**Fact 5.** *For affine classifiers of the form $f(\boldsymbol{x}) = \boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b}$, the subspace constrained perturbations are given as follows:*

$$\forall k \neq \hat{k}(\boldsymbol{x}_0), \boldsymbol{r}^k_{\mathcal{S}}(\boldsymbol{x}_0) = \frac{\left|f_k(\boldsymbol{x}_0) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0)\right|}{\|\mathbf{P}_{\mathcal{S}}\boldsymbol{w}_k - \mathbf{P}_{\mathcal{S}}\boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}\|_2^2}(\mathbf{P}_{\mathcal{S}}\boldsymbol{w}_k - \mathbf{P}_{\mathcal{S}}\boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}),$$

*where $\mathbf{P}_{\mathcal{S}}$ denotes the orthogonal projection operator onto subspace $\mathcal{S}$. The optimal subspace perturbation $\boldsymbol{r}^*_{\mathcal{S}}(\boldsymbol{x}_0)$ corresponds to the perturbation $\boldsymbol{r}^k_{\mathcal{S}}(\boldsymbol{x}_0)$ with minimal $\ell_2$ norm.*

*Proof.* We have, for $k \neq \hat{k}(\boldsymbol{x}_0)$,

$$\begin{aligned}\boldsymbol{r}^k_{\mathcal{S}} &= \underset{\boldsymbol{r}\in\mathcal{S}}{\operatorname{argmin}} \|\boldsymbol{r}\|_2^2 \text{ s.t. } f_k(\boldsymbol{x}_0+\boldsymbol{r}) \geq f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0+\boldsymbol{r}) \\ &= \underset{\boldsymbol{r}\in\mathcal{S}}{\operatorname{argmin}} \|\boldsymbol{r}\|_2^2 \text{ s.t. } (\boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)})^T(\boldsymbol{x}_0+\boldsymbol{r}) + b_k - b_{\hat{k}(\boldsymbol{x}_0)} \geq 0 \\ &= \underset{\boldsymbol{r}\in\mathbb{R}^d}{\operatorname{argmin}} \|\boldsymbol{r}\|_2^2 \text{ s.t. } (\boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)})^T(\boldsymbol{x}_0+\mathbf{P}_{\mathcal{S}}\boldsymbol{r}) + b_k - b_{\hat{k}(\boldsymbol{x}_0)} \geq 0.\end{aligned}$$

By equating the gradient of the Lagrangian of the above minimization problem to 0, we obtain

$$\boldsymbol{r}^k_{\mathcal{S}}(\boldsymbol{x}_0) = \frac{\lambda}{2}\mathbf{P}_{\mathcal{S}}(\boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}), \tag{5.14}$$

where $\lambda \geq 0$ denotes the Lagrangian multiplier. The complementary slackness condition reads

$$\lambda\left((\boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)})^T(\boldsymbol{x}_0+\mathbf{P}_{\mathcal{S}}\boldsymbol{r}) + b_k - b_{\hat{k}(\boldsymbol{x}_0)}\right) = 0.$$

If $\lambda = 0$, then $\boldsymbol{x}_0$ is exactly on the boundary, and we have $\boldsymbol{r}^k_{\mathcal{S}}(\boldsymbol{x}_0) = 0$. Otherwise, $\lambda > 0$, and we obtain

$$\lambda = 2\frac{f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0) - f_k(\boldsymbol{x}_0)}{\|\mathbf{P}_{\mathcal{S}}(\boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)})\|_2^2}.$$

By substituting in Eq. (5.14), we obtain the desired result. The fact that $\boldsymbol{r}^*_{\mathcal{S}}(\boldsymbol{x}_0)$ is then obtained as the minimum over all $\boldsymbol{r}^k_{\mathcal{S}}(\boldsymbol{x}_0)$ follows immediately from the argument in Fact 2. □

The resulting algorithm for computing the robustness to subspace-constrained perturbations is provided in Algorithm 3. The decision boundaries are iteratively linearized, and the update steps derived above are applied until the perturbed image is classified differently than the original one.

It is important to note that, while the (full) gradients of the classifications functions $f_k$ with respect to the input are required in order to compute the unconstrained worst-case perturbation (namely, when $\mathcal{S} = \mathbb{R}^d$), we only require the *projections* of the gradients onto the subspace $\mathcal{S}$ for estimating the subspace-constrained perturbation in Algorithm 3. In common cases where the dimension of the random subspace is $m \ll d$, the computation of the subspace constrained perturbation can therefore be more efficient, in addition to using less information about the classifier.

### 5.5.2 Experimental results

We now evaluate the robustness of different image classifiers to random and semi-random perturbations, and assess the accuracy of our bounds on various datasets and state-of-the-art

**Algorithm 3** Computing minimal perturbation in a subspace $\mathcal{S}$

**input:** Image $\boldsymbol{x}$, classifier $f$, orthogonal projector $\mathbf{P}_{\mathcal{S}}$ onto subspace $\mathcal{S}$.
**output:** Perturbation $\hat{\boldsymbol{r}}_{\mathcal{S}}$.
Initialize $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$, $i \leftarrow 0$.
**while** $\hat{k}(\boldsymbol{x}_i) = \hat{k}(\boldsymbol{x}_0)$ **do**
    **for** $k \neq \hat{k}(\boldsymbol{x}_0)$ **do**
        Compute the perturbation $\boldsymbol{r}^k_{\mathcal{S}}(\boldsymbol{x}_i)$ for the linearized classifier as follows:

$$\boldsymbol{r}^k_{\mathcal{S}}(\boldsymbol{x}_i) \leftarrow \frac{\left|f_k(\boldsymbol{x}_i) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)\right|}{\left\|\mathbf{P}_{\mathcal{S}}\left(\nabla f_k(\boldsymbol{x}_i) - \nabla f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)\right)\right\|_2^2} \mathbf{P}_{\mathcal{S}}\left(\nabla f_k(\boldsymbol{x}_i) - \nabla f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)\right),$$

    **end for**
    Set $\boldsymbol{r}_i$ to be the perturbation $\boldsymbol{r}^k_{\mathcal{S}}(\boldsymbol{x}_i)$ with minimal $\ell_2$ norm.
    Update the current datapoint: $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \boldsymbol{r}_i$
    $i \leftarrow i + 1$
**end while**
**return** $\hat{\boldsymbol{r}}_{\mathcal{S}} = \sum_i \boldsymbol{r}_i$

| | $m/d$ | | | | | |
|---|---|---|---|---|---|---|
| **Classifier** | 1 | $1/4$ | $1/16$ | $1/36$ | $1/64$ | $1/100$ |
| LeNet (MNIST) | 1.00 | $1.00 \pm 0.06$ | $1.01 \pm 0.12$ | $1.03 \pm 0.20$ | $1.01 \pm 0.26$ | $1.05 \pm 0.34$ |
| LeNet (CIFAR-10) | 1.00 | $1.01 \pm 0.03$ | $1.02 \pm 0.07$ | $1.04 \pm 0.10$ | $1.06 \pm 0.14$ | $1.10 \pm 0.19$ |
| VGG-F (ImageNet) | 1.00 | $1.00 \pm 0.01$ | $1.02 \pm 0.02$ | $1.03 \pm 0.04$ | $1.03 \pm 0.05$ | $1.04 \pm 0.06$ |
| VGG-19 (ImageNet) | 1.00 | $1.00 \pm 0.01$ | $1.02 \pm 0.03$ | $1.02 \pm 0.05$ | $1.03 \pm 0.06$ | $1.04 \pm 0.08$ |

Table 5.1: $\beta(f; m)$ for different classifiers $f$ and different subspace dimensions $m$. The VGG-F and VGG-19 are respectively introduced in [Cha+14; SZ14].

classifiers. Specifically, our theoretical results show that the robustness $\Delta_{\mathcal{S}}(\boldsymbol{x})$ of classifiers satisfying the curvature property precisely behaves as $\sqrt{d/m}\Delta_{\text{adv}}(\boldsymbol{x})$. We first check the accuracy of these results in different classification settings. For a given classifier $f$ and subspace dimension $m$, we define

$$\beta(f; m) = \sqrt{m/d}\frac{1}{|\mathscr{D}|}\sum_{\boldsymbol{x}\in\mathscr{D}}\frac{\Delta_{\mathcal{S}}(\boldsymbol{x})}{\Delta_{\text{adv}}(\boldsymbol{x})},$$

where $\mathcal{S}$ is chosen randomly for each sample $\boldsymbol{x}$ and $\mathscr{D}$ denotes the test set. This quantity provides indication to the accuracy of our $\sqrt{d/m}\Delta_{\text{adv}}(\boldsymbol{x})$ estimate of the robustness, and should ideally be equal to 1 (for sufficiently large $m$). Since $\beta$ is a random quantity (because of $\mathcal{S}$), we report both its mean and standard deviation for different networks in Table 5.1. For each network, we estimate the expectation by averaging $\beta(f; m)$ on 1000 random samples, with $\mathcal{S}$ also chosen randomly for each sample.

Observe that $\beta$ is suprisingly close to 1, even when $m$ is a small fraction of $d$. This shows that our quantitative analysis provide very accurate estimates of the robustness to semi-random noise. We visualize the robustness to random noise, semi-random noise (with

$m = 10$) and worst-case perturbations on a sample image in Figure 5.5. While random noise is clearly perceptible due to the $\sqrt{d} \approx 400$ factor, semi-random noise becomes much less perceptible even with a relatively small value of $m = 10$, thanks to the $1/\sqrt{m}$ factor that attenuates the required noise to misclassify the datapoint. It should be noted that the robustness of neural networks to adversarial perturbations has previously been observed empirically in [Sze+14], but we provide here a quantitative and generic explanation for this phenomenon. Note also that the generation of the semi-random perturbation in Fig. 5.5 (c) only required the projection of the gradients onto a random 10-dimensional subspace (as well as evaluations of $f$), whereas the generation of the adversarial example for Fig. 5.5 (d) required the knowledge of the full gradients of the network with respect to the input. This example shows that the full knowledge about the network (and in particular, its gradient) is not needed in order to find an imperceptible adversarial example that causes misclassification.

(a) (b) (c) (d)

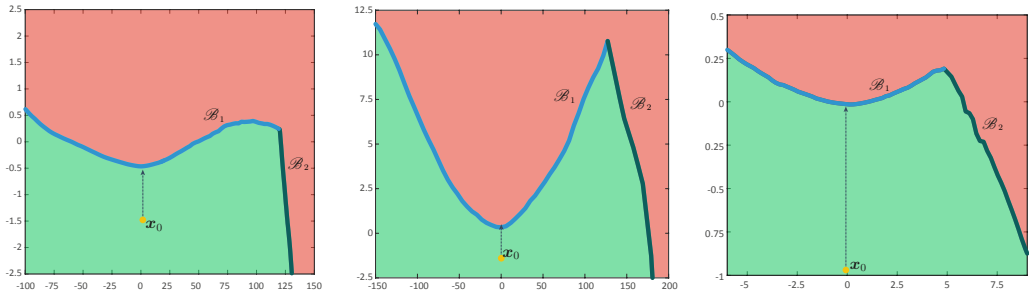
Figure 5.5: (a) Original image classified as "Cauliflower". Fooling perturbations for VGG-F network: (b) Random noise, (c) Semi-random perturbation with $m = 10$, (d) Worst-case perturbation, all wrongly classified as "Artichoke".

The high accuracy of our bounds for different state-of-the-art classifiers, and different datasets suggest that the decision boundaries of these classifiers have limited curvature $\kappa(\mathscr{B}_k)$, as this is a key assumption of our theoretical findings. To support the validity of this curvature hypothesis in practice, we visualize two-dimensional sections of the classifiers' boundary in Figure 5.6 in three different settings. Note that we have opted here for a visualization strategy rather than the numerical estimation of $\kappa(\mathscr{B})$, as the latter quantity is difficult to approximate in practice in high dimensional problems. In Figure 5.6, $\boldsymbol{x}_0$ is chosen randomly from the test set for each data set, and the decision boundaries are shown in the plane spanned by $\boldsymbol{r}^*$ and $\boldsymbol{r}^*_{\mathcal{S}}$, where $\mathcal{S}$ is a random *direction* (i.e., $m = 1$). Specifically, the following procedure is applied to sample from the boundary $\mathscr{B}_k$:

1. Choose a random direction, and estimate $\boldsymbol{r}^*_{\mathcal{S}}$ using Algorithm 3. Compute also the worst-case perturbation $\boldsymbol{r}^*$.
2. For each discretized value $\alpha_i \in [-T, T]$, do:
   (a) Define the datapoint $\boldsymbol{x}_i = \boldsymbol{x}_0 + \boldsymbol{r}^* + \alpha_i(\boldsymbol{r}^*_{\mathcal{S}} - \boldsymbol{r}^*)$.
   (b) Project the datapoint $\boldsymbol{x}_i$ onto the decision boundary using Algorithm 3: find the minimal subspace perturbation $\boldsymbol{r}^{2\mathrm{d}}_i \in \mathrm{span}(\boldsymbol{r}^*, \boldsymbol{r}^*_{\mathcal{S}})$ such that $\boldsymbol{x}_i + \boldsymbol{r}^{2\mathrm{d}}_i$ belongs to the decision boundary. Plot the projected point.

Different colors on the boundary correspond to boundaries with different classes. It can be observed that the curvature of the boundary is very small except at "junction" points

where the boundary of two different classes intersect. Our curvature assumption in Eq. (5.12), which only assumes a bound on the curvature of the decision boundary between pairs of classes $\hat{k}(\boldsymbol{x}_0)$ and $k$ (but not on the *global* decision boundary that contains junctions with high curvature) is therefore adequate to the decision boundaries of state-of-the-art classifiers according to Figure 5.6. Interestingly, the assumption in Corollary 1 is satisfied by taking $\kappa$ to be an empirical estimate of the curvature of the planar curves in Fig. 5.6 (a) for the dimension of the subspace being a *very* small fraction of $d$; e.g., $m = 10^{-3}d$. While not reflecting the curvature $\kappa(\mathscr{B}_k)$ that drives the assumption of our theoretical analysis, this result still seems to suggest that the curvature assumption holds in practice, and that the curvature of such classifiers is therefore very small. It should be noted that a related empirical observation was made in [GSS15]; our work however provides a precise quantitative analysis on the relation between the curvature and the robustness in the semi-random noise regime.




(a) VGG-F (ImageNet) (b) LeNet (CIFAR) (c) LeNet (MNIST)

Figure 5.6: Boundaries of three classifiers near randomly chosen samples. Axes are normalized by the corresponding $\Delta_{\text{adv}}$ since our assumption in the theoretical bound (Corollary 1) depends on the product of $\Delta_{\text{adv}}\kappa$. Note the difference in range between $x$ and $y$ axes. Note also that the range of horizontal axis in (c) is much smaller than the other two, hence the illustrated boundary is more curved.

We now show a simple demonstration of the vulnerability of classifiers to semi-random noise in Figure 5.7, where a structured message is hidden in the image and causes data misclassification. Specifically, we consider $\mathcal{S}$ to be the span of random translated and scaled versions of words "NIPS", "SPAIN" and "2016" in an image, such that $\lfloor d/m \rfloor = 228$. The resulting perturbations in the subspace are therefore linear combinations of these words with different intensities.[2] The perturbed image $\boldsymbol{x}_0 + \boldsymbol{r}^*_{\mathcal{S}}$ shown in Figure 5.7 (c) is clearly indistinguishable from Figure 5.7 (a). This shows that imperceptibly small structured messages can be added to an image causing data misclassification. This example might possibly lead to applications in automatic watermarking and steganography, where data is hidden in an image for tracking or to send hidden messages.

## 5.6 Conclusion

In this chapter, we precisely characterized the robustness of classifiers in a novel semi-random noise regime that generalizes the random noise regime. Specifically, our bounds relate

[2]This example departs somehow from the theoretical framework of this chapter, where *random* subspaces were considered. However, this empirical example suggests that the theoretical findings in this paper seem to approximately hold when the subspace $\mathcal{S}$ have statistics that are close to a random subspace.

(a) Image of a "Potflower"

(b) Structured perturbation containing random placement of words "NIPS", "2016", and "SPAIN"

(c) Classified as "Pineapple"

Figure 5.7: A fooling hidden message, $\mathcal{S}$ consists of linear combinations of random words.

the robustness in this regime to the robustness to adversarial perturbations. Our bounds depend on the *curvature* of the decision boundary, the data dimension, and the dimension of the subspace to which the perturbation belongs. Our results show, in particular, that when the decision boundary has a small curvature, classifiers are robust to random noise in high dimensional classification problems, even if the robustness to adversarial perturbations is small. Moreover, for semi-random noise that is mostly random and only mildly adversarial (i.e, the subspace dimension is small), our results show that state-of-the-art classifiers remain vulnerable to such perturbations. To improve the robustness to semi-random noise, our analysis encourages to impose geometric constraints on the curvature of the decision boundary, as we have shown the existence of an intimate relation between the robustness of classifiers and the curvature of the decision boundary.

In the following chapters, we focus on the study of the classifiers' robustness to structured geometric transformations of the images.

# 6 Quantifying invariance to geometric transformations

## 6.1 Introduction

In the previous chapters, we studied the robustness of classifiers to different types of perturbations in the data. The notion of robustness to perturbation is intimately related to that of *invariance*, which refers to the robustness of the classifier output to geometric transformations in the data (e.g., translation, rotations, etc...). The focus of this chapter is to study and *quantify* the invariance of classifiers. While the human visual system is invariant to some extent to geometric transformations, it is unclear whether automatic classifiers enjoy the same invariance properties. We propose in this chapter a principled and systematic method to measure the robustness of arbitrary image classifiers to geometric transformations. In particular, we design a new framework that can be applied to any low dimensional transformation group $\mathcal{T}$ (e.g., rotations, translations, ...) and to any classifier regardless of the particular nature of the classifier. For a given image, we define the invariance measure as the minimal distance between the identity transformation and a transformation in $\mathcal{T}$ that is sufficient to change the decision of the classifier on that image. In order to define the transformation metric, our novel key idea is to represent the set of transformed versions of an image as a manifold; the transformation metric is then naturally captured by the geodesic distance on the manifold. Hence, for a given image, our invariance measure essentially corresponds to the minimal geodesic distance on the manifold that leads to a point where the classifier's decision is changed. A global invariance measure is then derived by averaging over a sufficiently large sample set. Equipped with our generic definition of invariance, we leverage the techniques used in the analysis of manifolds of transformed visual patterns [DG05; Wak+05; JDV08] and design a method built on the efficient Fast Marching algorithm [KS98; Tsi95] to compute the invariance of classifiers.

Using the proposed method, we quantify the invariance of several classifiers, and provide a comparative study of classifiers with respect to their invariance to geometric transformations. We show in particular that convolutional neural network (CNN) architectures and classifiers based on scattering transforms [BM13] have a larger invariance score when compared to other classifiers, such as SVM classifiers. We highlight moreover that the invariance of convolutional neural networks increases with the network *depth*. While this was previously known to hold in simple settings where toy images and simple transformation sets are

Part of this chapter has been published in [FF15].

considered, our approach provides a quantitative and systematic method to verify this in more complex settings. Our proposed method moreover quantitatively demonstrates that convolutional neural networks (CNN) classifiers are not sufficiently invariant to small combinations of translations, rotations and dilations of the image, despite the common belief that these classifiers have good invariance properties. To improve the invariance properties of classifiers, we show that data augmentation, where transformed images are added to the training set, can be a very effective solution. By providing a systematic tool to assess the classifiers in terms of their robustness to geometric transformations, we bridge a gap towards understanding the invariance properties of different families of classifiers.

The chapter is organized as follows: in Section 6.2, we propose a mathematical formulation of the problem of quantifying the invariance of a classifier to geometric transformations. In Section 6.3, we propose an algorithm for computing this quantity, and we perform experiments on the invariance of classifiers in Section 6.4. We finally conclude in Section 6.5.

## 6.2 Problem formulation

### 6.2.1 Definitions

We consider a mathematical model where images are represented as functions $x : \mathbb{R}^2 \to \mathbb{R}$, and we denote by $L^2$ the space of square integrable images. Let $\mathcal{T}$ be a Lie group consisting of geometric transformations on $\mathbb{R}^2$, and we denote by $p$ the dimension of $\mathcal{T}$ (i.e., number of free parameters), which is assumed to be sufficiently small in this chapter (i.e., $p \leq 6$). For any transformation $\tau$ that belongs to $\mathcal{T}$, we denote by $x_\tau$ the image $x$ transformed by $\tau$. That is, $x_\tau(u_1, u_2) = x(\tau^{-1}(u_1, u_2))$. Examples of Lie groups include the rotation group $\mathrm{SO}(2)$ ($p = 1$, described by one angle) and the similarity group ($p = 4$, described by a 2D translation vector, a dilation and an angle).

Consider an image classification task, where the images are assigned discrete labels in $\mathcal{L} = \{1, \ldots, L\}$, and let $\hat{k}$ be an arbitrary image classifier. Formally, $\hat{k}$ is a function defined on the space of square integrable images $L^2$, and takes values in the set $\mathcal{L}$. Our goal is to evaluate the invariance of $\hat{k}$ with respect to $\mathcal{T}$. Given an image $x$, we define the invariance score of $\hat{k}$ relative to $x$, $\Delta_\mathcal{T}(x)$[1], to be the *minimal normalized distance* from the identity transformation to a transformation $\tau$ that changes the classification label, i.e.,

$$\Delta_\mathcal{T}(x) = \min_{\tau \in \mathcal{T}} \frac{d(e, \tau)}{\|x\|_{L^2}} \text{ subject to } \hat{k}(x_\tau) \neq \hat{k}(x), \tag{6.1}$$

where $e$ is the identity element of the group $\mathcal{T}$ and $d : \mathcal{T} \times \mathcal{T} \to \mathbb{R}^+$ is a metric on $\mathcal{T}$ that we define later (Section 6.2.2). The invariance score quantifies the resilience of the classifier to transformations in $\mathcal{T}$; larger values of $\Delta_\mathcal{T}(x)$ indicate a larger invariance. It is worth noting that the definition of $\Delta_\mathcal{T}$ is tightly related to the definition of adversarial perturbations in Chapter 3. Specifically, we consider here *geometric* perturbations (instead of arbitrary additive perturbations) and focus on the minimal such transformation (in the

[1]We remove the explicit dependence on the classifier $\hat{k}$ to simplify notations. The classifier will be clear from the context.

sense of a metric defined on $\mathcal{T}$).

Finally, for a given distribution of data points $\mu$, the global invariance score of the classifier $\hat{k}$ to transformations in $\mathcal{T}$ is defined by

$$\rho_{\mathcal{T}}(\hat{k}) = \mathbb{E}_{x\sim\mu}\Delta_{\mathcal{T}}(x). \tag{6.2}$$

The quantity $\rho_{\mathcal{T}}(\hat{k})$ depends on $\hat{k}$ as well as the distribution of data points $\mu$. To simplify notations, we have omitted the dependence on $\mu$, assuming the distribution is clear from the context. We estimate in practice the global invariance by taking the empirical average[2] over training points: $\hat{\rho}_{\mathcal{T}}(\hat{k}) = \frac{1}{m}\sum_{j=1}^{m}\Delta_{\mathcal{T}}(x_j;\hat{k})$.

### 6.2.2 Transformation metric

We discuss and introduce the distance used for the invariance score $\Delta_{\mathcal{T}}(x)$. It should be noted that $\mathcal{T}$ is possibly a multi-dimensional group (i.e., the transformations in $\mathcal{T}$ are described by several parameters of different nature such as translation, rotation, scale, ...); hence, defining a trivial metric that measures the absolute distance between transformation parameters is of limited interest, as it combines parameters possibly of different nature. Instead, a more relevant notion of distance is one that *depends on the underlying image* $x$. In that case, the distance $d(\tau_1,\tau_2)$ quantifies the change in *appearance* between images $x_{\tau_1}$ and $x_{\tau_2}$, rather than an absolute distance between the two transformations. Consider for example the *image distance* $d_x(\tau_1,\tau_2) = \|x_{\tau_1} - x_{\tau_2}\|_{L^2}$. While $d_x$ explicitly depends on the underlying image $x$, it fails to capture the intrinsic geometry of the family of transformed images. To illustrate this point, we consider a simple example of images in Fig. 6.1 with two transformed versions $x_{\tau_1}$ and $x_{\tau_2}$ of a reference image $x_{\tau_0}$. Note that $d_x(\tau_0,\tau_1) = d_x(\tau_0,\tau_2)$, as both transformed objects ($x_{\tau_1}$ and $x_{\tau_2}$) have no intersection with the reference object ($x_{\tau_0}$). However, it is clear that $x_{\tau_2}$ incurred a large rotation and translation, while $x_{\tau_1}$ underwent a slight vertical translation. Hence, the distance metric should naturally satisfy $d(\tau_0,\tau_1) < d(\tau_0,\tau_2)$, which is not the case for the image distance. This is crucial in our setting, as a classifier that recognizes the similarity of the objects in $x_{\tau_2}$ and $x_{\tau_0}$ is certainly more robust to transformations than a classifier that merely recognizes the similarity between $x_{\tau_1}$ and $x_{\tau_0}$, and should be given a higher score. This example underlines a well-known fundamental issue with the $L^2$ distance that fails to capture the intrinsic distance of the curved manifold of transformed images (see e.g., [TDSL00; DG05]). To correctly capture the intrinsic structure of the manifold, we define $d$ to be the length of the shortest path belonging to the manifold (i.e., the *geodesic distance*). For illustration, we show in Fig. 6.2 images along the geodesic path from $x_{\tau_0}$ to $x_{\tau_2}$; the geodesic distance is then essentially the sum of *local* $L^2$ distances between transformed images over the geodesic path. We formalize these notions as follows.

Let $\mathcal{M}(x)$ be the family of transformed images $\mathcal{M}(x) = \{x_\tau : \tau \in \mathcal{T}\}$. Equipped with the $L^2$ metric, $\mathcal{M}(x)$ defines a metric space and a continuous submanifold of $L^2$. Following the works of [Wak+05] and [JDV08] that have considered similar manifolds in different contexts, we call $\mathcal{M}(x)$ an *Image Appearance Manifold* (IAM), and we follow here their approach. Assuming that $\gamma : [0,1] \mapsto \mathcal{T}$ is a $C^1$ curve in $\mathcal{T}$, and that $x_{\gamma(t)}$ is differentiable

[2] In practice, it is sufficient to consider an empirical average over a sufficiently large random subset of the training set. The number of samples is chosen to achieve a small enough confidence interval.

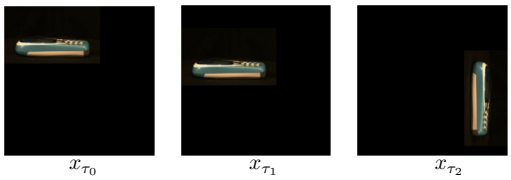



Figure 6.1: Schematic representation of the problem encountered by using metric the $L^2$ metric. Black pixels indicate pixels with value 0, and $x_{\tau_1}, x_{\tau_2}$ are obtained by applying a combination of rotation and translation to $x_{\tau_0}$. Original image taken from [GBS05].

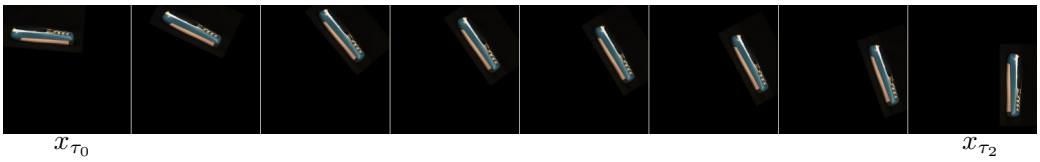



Figure 6.2: Images along the geodesic path from $x_{\tau_0}$ to $x_{\tau_2}$. Original image taken from [GBS05].

with respect to $t$, we define the *length* $L(\gamma)$ of $\gamma$ as

$$L(\gamma) = \int_0^1 \left\| \frac{d}{dt} x_{\gamma(t)} \right\|_{L^2} dt. \tag{6.3}$$

Note that Eq. (6.3) is expressed in terms of the $L^2$ metric in the image appearance manifold and corresponds to summing the local $L^2$ distances between transformed images over the path $x_\gamma$. We now show that $L(\gamma)$ can be expressed as a length associated to a Riemannian metric on $\mathcal{T}$ that we now derive. Defining the map

$$F : \mathcal{T} \to \mathcal{M}, \quad \tau \mapsto x_\tau,$$

we have

$$\frac{d}{dt} x_{\gamma(t)} = (F \circ \gamma)'(t) = dF_{\gamma(t)}(\gamma'(t)),$$

where $dF_\tau$ denotes the differential of $F$ at $\tau$, and $\gamma'$ is derivative of $\gamma$. It follows that

$$L(\gamma) = \int_0^1 \sqrt{g_{\gamma(t)}(\gamma'(t), \gamma'(t))} dt$$

where $g_\tau$ is the *Riemannian metric* (i.e., a positive bilinear form on $T_\tau\mathcal{T}$, the tangent space of $\mathcal{T}$ at $\tau$), given by:

$$g_\tau(v, w) = \langle dF_\tau(v), dF_\tau(w) \rangle_{L^2} \text{ for all } v, w \in T_\tau\mathcal{T}.$$

Note that $g$ can be equivalently seen as the pullback of the $L^2$ metric on $\mathcal{M}(x)$ along $F$. By choosing a basis in the tangent space, the length $L(\gamma)$ can be equivalently written

$$L(\gamma) = \int_0^1 \sqrt{\gamma'(t)^T G_{\gamma(t)} \gamma'(t)} dt,$$

where $G_{\gamma(t)}$ is the $p \times p$ positive definite matrix associated to the bilinear form $g$.

Having defined the length of a curve on $\mathcal{T}$, the geodesic distance between two points $\tau_1, \tau_2$ is defined as the length of the shortest curve joining the two points:

$$d(\tau_1, \tau_2) = \inf\{L(\gamma) : \gamma \in C^1([0,1]), \gamma(0) = \tau_1, \gamma(1) = \tau_2\}.$$

Finally, our problem therefore consists in computing the global invariance score, or equivalently $\Delta_{\mathcal{T}}(x)$ defined in Eq. (6.1), where $d$ is the geodesic distance. In other words, our problem becomes that of computing the minimal geodesic distance from the identity transformation to a transformation that is sufficient to change the estimated label of $\hat{k}$.

## 6.3 Invariance score computation

We now propose a generic method for computing the invariance score $\Delta_{\mathcal{T}}(x)$. The key to an efficient and accurate approximation of $\Delta_{\mathcal{T}}(x)$ lies in the effective computation of geodesics on the manifold $(\mathcal{T}, G)$ that we address as follows.

Let $u(\tau) = d(e, \tau)$ be the *geodesic map* that measures the geodesic distance between the (fixed) identity element and $\tau$. The geodesic map satisfies the following Eikonal equation [Pey+10]

$$\|\nabla u(\tau)\|_{G_\tau^{-1}} = 1 \text{ for } \tau \in \mathcal{T} \backslash \{e\}, \text{ and } u(e) = 0, \tag{6.4}$$

where $\|x\|_A = \sqrt{\langle x, x \rangle_A}$ with $\langle x, y \rangle_A = x^T A y$. Moreover, it was proved in [CL83] that the geodesic map $u$ is the *unique* viscosity solution of the Eikonal equation, provided that $\tau \to G(\tau)$ is continuous. Many numerical schemes rely on the Eikonal equation characterization to approximate the geodesic map. We use here the popular *Fast Marching (FM) method* [KS98], a fast front propagation approach that computes the values of the discrete geodesic map in increasing order. To simplify the exposition of FM, we focus here on the case where the manifold $\mathcal{T}$ is two-dimensional (i.e., $p = 2$). The extension to arbitrary dimensions is straightforward, and we refer to [Pey+10; SV03] for more complete explanations and computations.

We assume that the manifold $\mathcal{T}$ is sampled using a regular grid; let $\mathcal{T}_*$ be the sampling of $\mathcal{T}$, and $U$ be the discrete vector that approximates $u$ at the nodes. The structure of Fast Marching is almost identical to Dijkstra's algorithm for computing shortest paths on graphs [Dij59]. The main difference lies in the update step, which bypasses the constraint of propagation along edges. For a given node $\tau$, define $\mathcal{N}(\tau)$ to be the set of neighbours of $\tau$ (see illustration in Fig. 6.3).

In the FM algorithm, each grid point is tagged either as *Known* (nodes for which distance is frozen), or *Unknown* (nodes for which distance can change in subsequent iterations). Initially, the grid points are set to *Unknown*, and $U$ is set to $\infty$, except $U(e)$ that is set to zero. At each iteration of FM, the unknown node $\tau_{\min}$ with smallest $U$ is selected, and tagged as *Known*. Then, each unknown neighbour $\tau \in \mathcal{N}(\tau_{\min})$ is visited, and $U(\tau)$ is

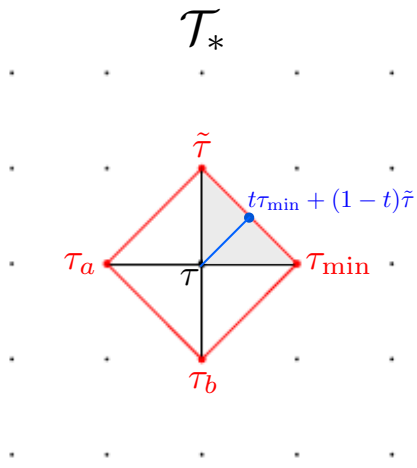



Figure 6.3: Schematic representation of the discretized manifold $\mathcal{T}_*$, and the Fast Marching update rule. In this figure, we have $\mathcal{N}(\tau) = \{\tilde{\tau}, \tau_{\min}, \tau_a, \tau_b\}$.

**Algorithm 4** Computation of the invariance score $\Delta_{\mathcal{T}}(x)$ (for $p = 2$)

Initialize $U(e) = 0$, $U = \infty$ otherwise, and tag all nodes as *unknown*.
**while** termination criterion is not met **do**
  Select the *unknown* node $\tau_{\min}$ that achieves minimal distance $U$.
  Tag $\tau_{\min}$ as *known*.
  If $\hat{k}(x_{\tau_{\min}}) \neq \hat{k}(x)$, set $\Delta_{\mathcal{T}}(x; \hat{k}) \leftarrow U(\tau_{\min})/\|x\|_{L^2}$ and terminate.
  **for all** *unknown* $\tau \in \mathcal{N}(\tau_{\min})$ **do**
    Update $U(\tau)$ to be the minimum of itself, $U(\tau_{\min}) + \|\tau - \tau_{\min}\|_{G_\tau}$ and the expression in Eq.(6.5).
  **end for**
**end while**

updated as follows: $U(\tau)$ is set to be the minimum of itself, $U(\tau_{\min}) + \|\tau - \tau_{\min}\|_{G_\tau}$ and

$$\min_{t \in [0,1]} tU(\tau_{\min}) + (1-t)U(\tilde{\tau}) + \|t\tau_{\min} + (1-t)\tilde{\tau} - \tau\|_{G_\tau}, \tag{6.5}$$

for each known $\tilde{\tau}$ such that $(\tau, \tau_{\min}, \tilde{\tau})$ forms a triangle (see Fig. 6.3). It is worth noting that, unlike Dijkstra, FM seeks the optimal point (possibly outside the set $\mathcal{T}_*$) on the neighbourhood boundary that minimizes the estimated distance at $\tau$, under a linear approximation assumption (Eq. (6.5)). Fortunately, the problem in Eq. (6.5) can be solved in closed form, as it corresponds to the minimization of a scalar quadratic equation [SV03].

The proposed method, which uses FM algorithm to compute $\Delta_{\mathcal{T}}(x)$, is given in Algorithm 4 in the two dimensional case. The algorithm is stopped whenever a transformation that changes the classification label is found.[3] The nodes and metrics are generated on-the-fly in order to avoid spending unnecessary ressources on far-away nodes that might be farther than the minimal transformation that satisfies $\hat{k}(x) \neq \hat{k}(x_\tau)$ and therefore never visited.

[3]To ensure the termination of the algorithm (even if no successful transformation is found) we limit the number of iterations $N$ to $50,000$. However, in all our experiments, this limit was never reached, and the algorithm terminated by successfully finding a transformation that satisfies $\hat{k}(x_\tau) \neq \hat{k}(x)$.

The complexity of the proposed method is $O(N \log(N))$, where $N$ is the number of visited nodes if a min-heap structure is used (for constant $p$, and constant cost for evaluation of $\hat{k}$). It is important to note however that the complexity of the algorithm has an exponential dependence on the dimension $p$ since our method involves the enumeration of simplices in dimension $p$; this is however not a big limitation as our main focus goes to low-dimensional transformation groups.

Finally, we note that when the metric is isotropic (i.e., $G_\tau$ is proportional to the identity matrix for all $\tau$), FM provides a consistent scheme [Pey+10]. That is, as the discretization step tends to zero, the solution computed by the algorithm tends towards the viscosity solution of the Eikonal equation. Unfortunately, for arbitrary anisotropic metrics, consistency is however not guaranteed, and the exact computation of the geodesics becomes much more difficult and computationally demanding (see [SV00; BC11; Lin03; Mir14]). We empirically observed that the anisotropy of the considered metric is generally not very large in the vicinity of $e$ (although it exceeds the theoretical limit of guaranteed consistency), which leads to approximately accurate estimates of the geodesic distance using the proposed method, when the discretization step is sufficiently small. It should finally be noted that all previous methods addressing the metric anisotropy can readily be applied to our setting. At the expense of higher computational cost, these methods provide theoretical guarantees on the accuracy of the geodesic distance estimation.

## 6.4 Experiments: analysis of the invariance of classifiers

We propose now a set of experiments to study the invariance of classifiers in different settings. In particular, we consider the following transformation groups:

- $\mathcal{T}_{\text{trans}}$: in-plane translations of the image ($p = 2$),
- $\mathcal{T}_{\text{dil+rot}}$: dilations and rotations around the center of the image ($p = 2$),
- $\mathcal{T}_{\text{sim}}$: similarity transformations that describe combinations of translations, dilations and rotations around the center of the image ($p = 4$).

In all experiments, we used a discretization step of 0.5 pixels for translations, $\pi/20$ radians for rotation, and 0.1 for dilation for the proposed method. Finally, the transformed images have the same size as the original image, and we use a zero-padding boundary condition.

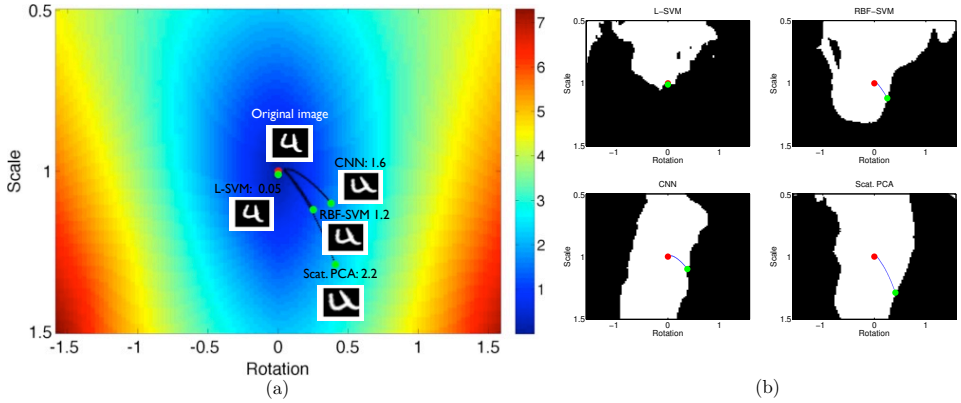
### 6.4.1 Evaluation of invariance on MNIST handwritten digits dataset

We first compare the invariance of different classifiers on the MNIST handwritten digits dataset [LeC+98a]. We consider the following classifiers:

1. **Linear SVM** [Fan+08b],
2. **SVM with RBF kernel** [CL11],
3. **Convolutional Neural Network** [VL15]: we employ a baseline architecture with two hidden layers containing each a convolution operation ($5 \times 5$ filters with 32 feature maps for the first layer and 64 for the second layer), a rectified linear unit nonlinearity, and a max pooling over $2 \times 2$ windows followed by a subsampling. The architecture is trained with stochastic gradient descent, with a softmax loss.
4. **Scattering transform followed by a generative PCA classifier**. The same

| Group | L-SVM | RBF-SVM | CNN | Scat. PCA |
|---|---|---|---|---|
| Test error (%) | 8.4 | 1.4 | **0.7** | 0.8 |
| Translations ($\mathcal{T} = \mathcal{T}_{\text{trans}}$) | 0.8 | 1.3 | 1.7 | **2.1** |
| Dilations + Rotations ($\mathcal{T} = \mathcal{T}_{\text{dil+rot}}$) | 0.8 | 1.5 | **1.9** | 1.8 |
| Similarity ($\mathcal{T} = \mathcal{T}_{\text{sim}}$) | 0.6 | 1.1 | 1.5 | **1.6** |

Table 6.1: Accuracy and invariance scores of different classifers on the MNIST dataset.




(a) (b)

Figure 6.4: Distance map with $\mathcal{T}_{\text{dil+rot}}$ group (left), and correctly classified regions (right), for the four tested classifiers on a "4" digit image. *Details for a):* the color code indicates the geodesic distance from the identity transformation (shown by red dot at the center). For each classifier, the minimal transformation for which the output of the classifier is not correct (i.e., not "4") is indicated, along with the corresponding transformed image and geodesic path. *Details for b):* the region where the classifier correctly outputs the label "4" is shown in white. Geodesic paths are also shown.

settings as in the MNIST handwritten classification example [BM13] were used. In particular, the features are obtained using a two-level scattering operator, with a spatial size window parameter determined using a cross-validation procedure. A generative PCA classifier is then trained by fitting an affine subspace to the scattering features of each class, and a label is determined at test time by selecting the subspace index with smallest error norm when projecting onto the affine subspaces.

Table 6.1 reports the performance of the different classifiers under study, and their invariance scores $\hat{\rho}_{\mathcal{T}}(\hat{k})$ using the proposed method. As expected, the linear and RBF-SVM classifiers compare poorly to other classifiers in terms of invariance. This is due to the construction of the CNN and Scat. PCA, which explicitly take into account the invariance. Moreover, it can be noted that Scat. PCA outperforms CNN in terms of robustness to translations, and global similarity transformations, even if the two classifiers have similar test errors. This result is in agreement with the theoretical evidence in [BM13; Mal12] showing that scattering representations are invariant to deformations. It should be noted however that the notion of invariance developed in the latter papers is concerned with the *feature* representation step rather than the *overall* classification architecture (i.e., mapping from the input to label; hence including feature representation *and* classification), which is measured by our invariance score.

To get further insights on the invariance of the classifiers, we focus on the two-dimensional group $\mathcal{T}_{\text{dil+rot}}$, and show in Fig. 6.4 (a) the geodesic distance map for an example image of digit "4" computed starting from the identity transformation (shown by a red dot at the center). Moreover, we overlay the minimally transformed images that change the labels of each of the classifiers, along with the corresponding geodesic paths. On this example, the Scat. PCA classifier is the most robust: a large dilation, accompanied with a rotation is required to change the classification label. In contrast, the linear SVM is easily "fooled" with a slight dilation. In Fig. 6.4 (b) we illustrate in white the region of the Rotation-Scale plane, where the classifier outputs the correct label "4". Interestingly, the CNN and Scat. PCA classifiers are largely invariant to dilations (indicated by the vertical shape of the white region), while being moderately robust to rotations.

The geodesic distance map strongly depends on the considered image. To show this point, we do the same experiment as above, but using a "0" digit image and show the results in Fig. 6.5. Observe that the distance maps have very different shapes, which can be explained by the fact that the digit "0" only changes slightly as a rotation is applied, while the appearance of digit "4" is strongly affected by rotation. Also, observe that CNN and scattering classifiers are robust to rotations in this example, and a dilation is needed in order to switch the label of the original image in this example.

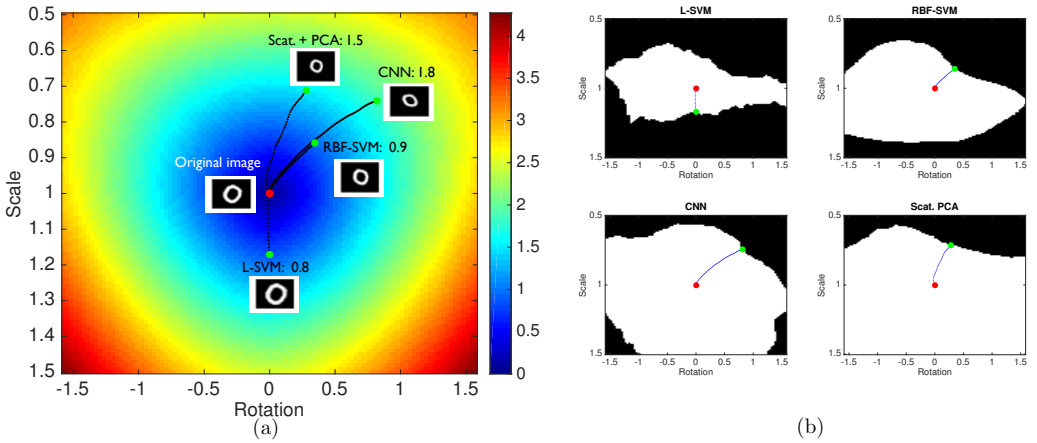
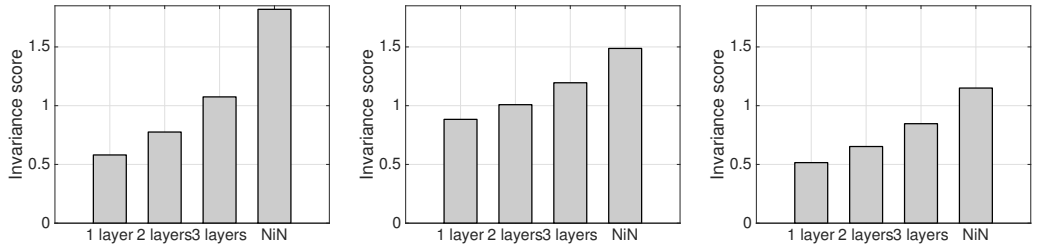



Figure 6.5: Distance map with $\mathcal{T}_{\text{dil+rot}}$ group (left), and correctly classified regions (right), for the four tested classifiers on a "0" digit image. *Details for a):* the color code indicates the geodesic distance from the identity transformation (shown by red dot at the center). For each classifier, the minimal transformation for which the output of the classifier is not correct (i.e., not "0") is indicated, along with the corresponding transformed image and geodesic path. *Details for b):* the region where the classifier correctly outputs the label "0" is shown in white. Geodesic paths are also shown.

### 6.4.2 Evaluation of invariance on CIFAR-10 natural images dataset

In this second experimental section, we perform experiments on the CIFAR-10 dataset [KH09]. We focus on baseline CNN classifiers, and learn architectures with 1, 2 and 3 hidden layers. Specifically, each layer consists of a successive combination of convolutional, rectified linear units and pooling operations. The convolutional layers consist of $5 \times 5$ filters

with respectively $32, 32$ and $64$ feature maps for each layer, and the pooling operations are done on a window of size $3 \times 3$ with a stride parameter of 2. We build the three architectures gradually, by successively stacking a new hidden layer on top of the previous architecture (kept fixed). The last hidden layer is then connected to a fully connected layer, and the softmax loss is used. Moreover, the different architectures are trained with stochastic gradient descent. On the test set, the error of the architectures with 1, 2 and 3 layers are respectively 35.6%, 25.0% and 22.7%. For completeness, we also consider a Network-in-Network deep neural network [LCY14] that is known to achieve very good results on the CIFAR-10 dataset. Specifically, the network is composed of 3 micro-networks, containing each a convolutional layer, rectified nonlinearity, a multilayer perceptron with 2 hidden layers, and a max-pooling. Compared to the other tested architectures, the NiN deep net is therefore a significantly more complex architecture. We train this neural network (without pre-processing), and obtain an error rate of 11.7 %.




(a) Translations (b) Dilation + Rotation (c) Similarity

Figure 6.6: Invariance scores of CNNs on $\mathcal{T}_{\text{trans}}$, $\mathcal{T}_{\text{dil+rot}}$ and $\mathcal{T}_{\text{sim}}$, for the CIFAR-10 dataset.

We show in Fig. 6.6 the invariance scores of the different architectures. Our approach captures the *increasing* invariance with the number of layers of the network, for the three groups under study. This result is in agreement with empirical studies and previous known belief [Goo+09; BCV13] that invariance increases with the depth of the network. However, while previous results were measuring the invariance with respect to a one dimensional transformation group (e.g., rotation only), the proposed method provides a systematic and principled way of verifying the increased invariance of CNNs with depth on more complex groups (e.g., similarity transformations). Interestingly enough, it should be noted that despite the relatively small difference in performance between the two and three layers architectures, the invariance score strongly increases. This highlights once again that invariance and test error capture two different properties of classifiers.

We further visualize the invariance level of the NiN network on the CIFAR-10 dataset and show in Fig. 6.7 some sample images from the training set sorted according to their invariance score $\Delta_{\mathcal{T}}$ for the similarity group. It should be noted that despite the high accuracy achieved by the NiN network, the distinction between the transformed and original images is hardly perceptible for most sampled images; a slight transformation of the image is therefore sufficient to change the label of the NiN classifier. For the top-scored images however, the difference between the original and the minimally transformed images is clearly perceptible even though a human observer is likely to correctly recognize the class of the transformed images. For completeness, we perform the same experiment on the translations group (i.e., $\mathcal{T} = \mathcal{T}_{\text{trans}}$) and illustrate the sorted images in Fig. 6.8. It can be observed that the network is much less robust to *combinations* of 2D translations, rotation and dilation,

than to translation alone. In fact, note that the difference between the transformed images and original images in Fig. 6.8 is more pronounced than in Fig. 6.7. In the following section, we quantify the effect of *data augmentation*, a widely used technique in classification for improving the accuracy and invariance.

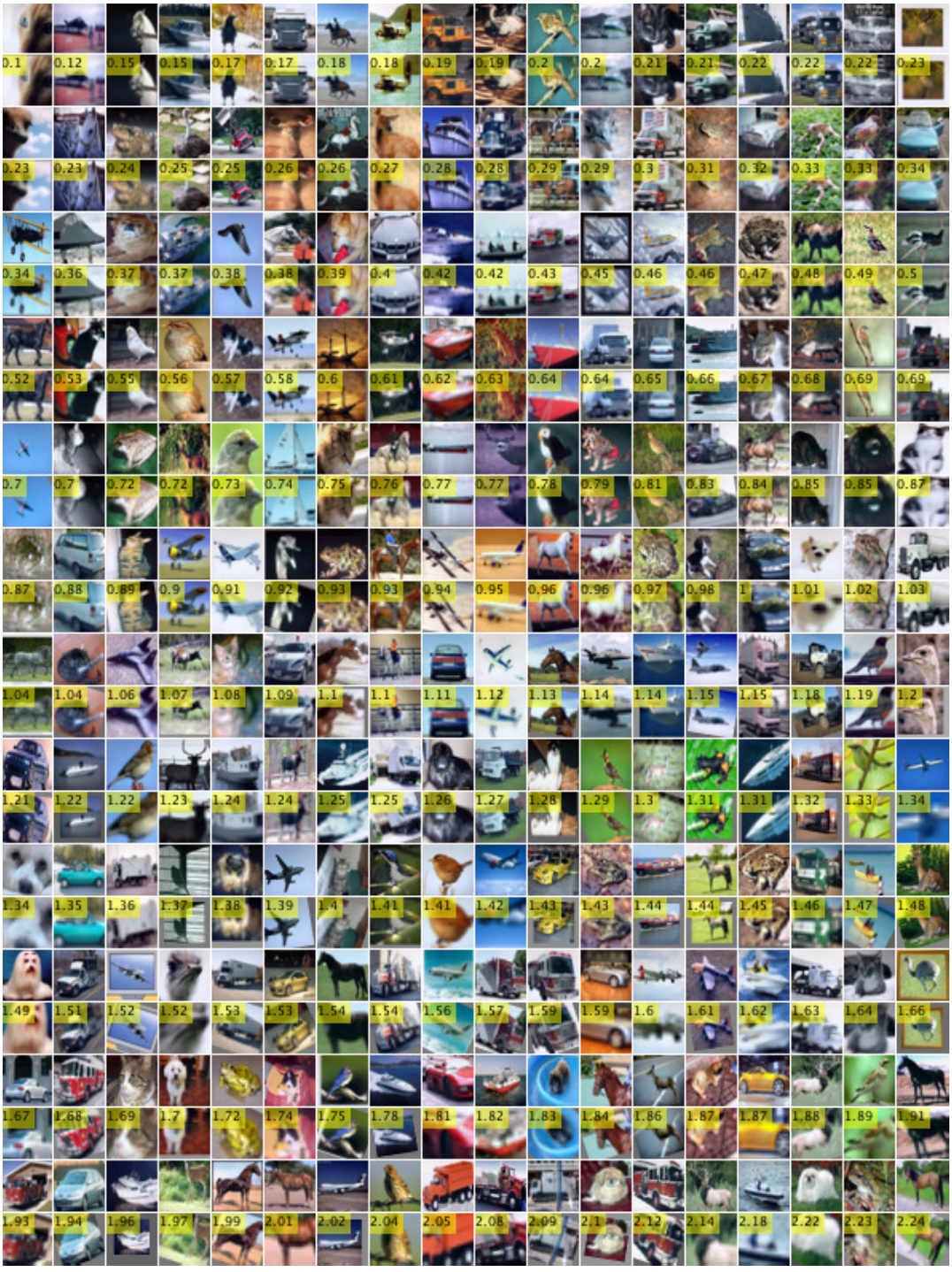


Figure 6.7: Sample images from the CIFAR-10 dataset and their invariance to similarity transformations $\Delta_{\mathcal{T}}(x)$ (with $\mathcal{T} = \mathcal{T}_{\text{sim}}$) for the NiN classifier. The odd rows show the original images, and the even rows show the minimally transformed images changing the prediction of the classifier. The invariance score $\Delta_{\mathcal{T}}(x)$ is indicated on each transformed image. All original images are **correctly classified** by the NiN classifier. We have $\rho_{\mathcal{T}}(\hat{k}) = 1.15$.

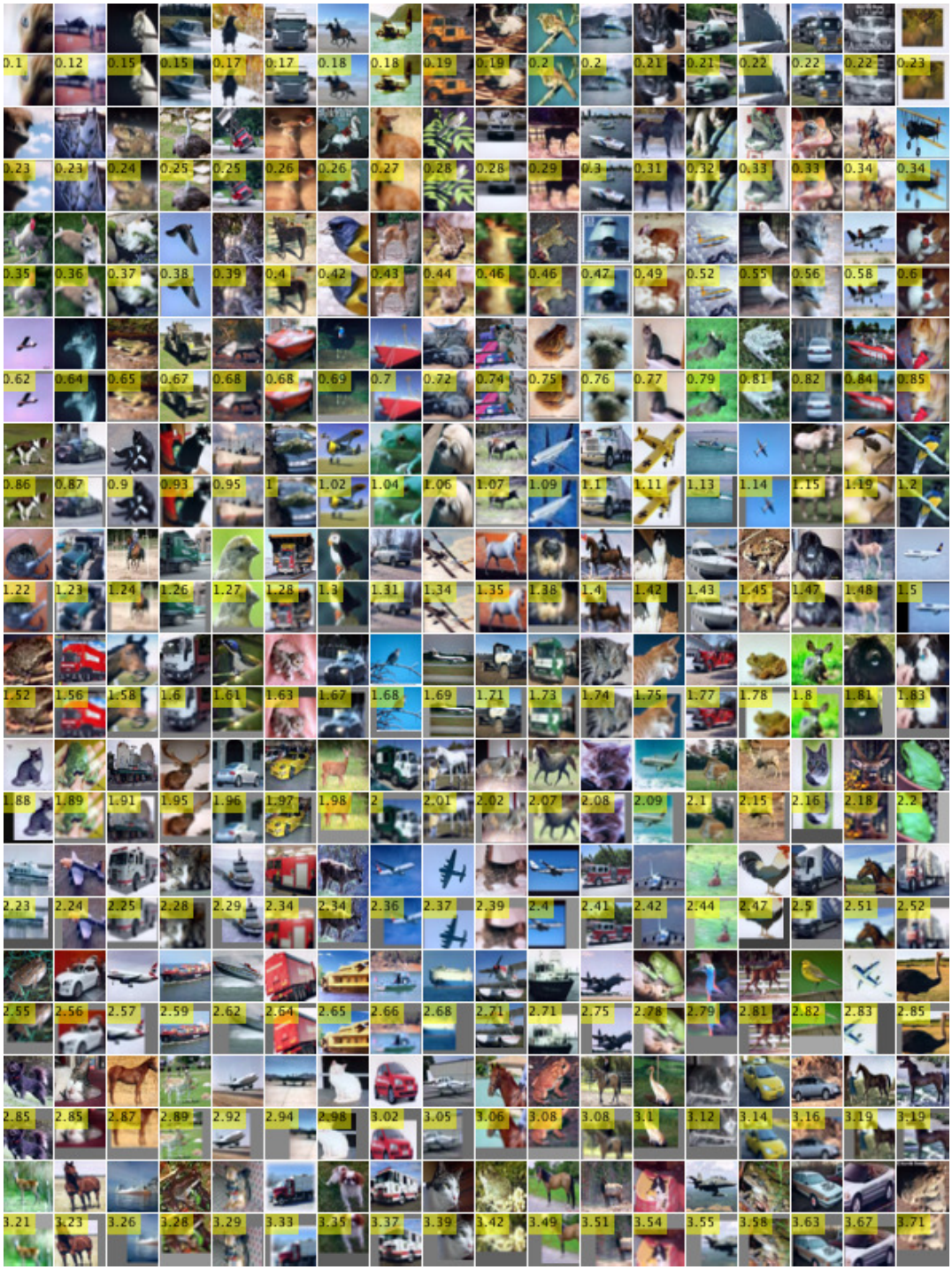
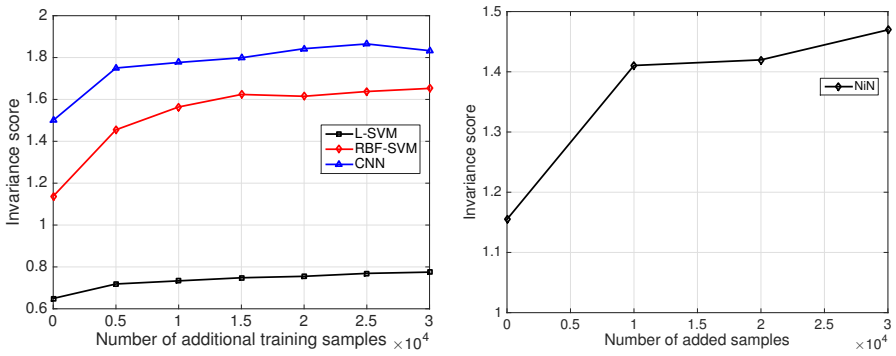


Figure 6.8: Sample images from the CIFAR-10 dataset and their invariance to translation $\Delta_{\mathcal{T}}(x)$ (with $\mathcal{T} = \mathcal{T}_{\text{trans}}$) for the NiN classifier. The odd rows show the original images, and the even rows show the minimally transformed images changing the prediction of the classifier. The invariance score $\Delta_{\mathcal{T}}(x)$ is indicated on each transformed image. All original images are **correctly classified** by the NiN classifier. We have $\rho_{\mathcal{T}}(\hat{k}) = 1.82$.

### 6.4.3 Effect of data augmentation on the invariance

In vision tasks, it is common practice to augment the training data with artificial examples obtained by slightly distorting the original examples to achieve invariance. Although this practice is known to improve the classification performance of the classifiers on many tasks, its effect on the invariance of the classifier is not quantitatively understood. We quantify in this section the effect of data augmentation on the invariance by training classifiers on augmented training sets, where randomly selected images from the training set undergo random transformations[4] from the similarity group $\mathcal{T}_{\text{sim}}$, and are then added to the training set. We perform experiments on the MNIST task, where L-SVM, RBF-SVM and CNN classifiers (introduced in Section 6.4.1) are used, as well as the CIFAR-10 task, where the NiN classifier (introduced in Section 6.4.2) is used. Fig. 6.9 shows the invariance score with respect to the number of added transformed samples for the different classifiers on the two tasks. Note that all classifiers improve their invariance score as more transformed samples are added to the training set. Moreover, the RBF-SVM improves its invariance score by around 50% with mere additions of artificial examples in the training set, and outperforms the invariance of CNN (without data augmentation). Interestingly, the invariance score of the augmented RBF-SVM classifier is comparable to Scat. PCA classifier, which is carefully designed to satisfy invariance properties. Observe finally that data augmentation similarly improves the invariance score of CNN and NiN classifiers substantially on the MNIST and CIFAR-10 classification tasks. We further illustrate this point qualitatively by comparing minimally transformed images that change the estimated label of the classifier for classifiers that are trained with and without data augmentation. The results are shown in Fig. 6.10. Note that the minimal transformations required to change the label are substantially larger for networks that are trained with data augmentation than without data augmentation. This provides a qualitative confirmation to the results in Fig. 6.9. This experiment highlights the power of a simple technique, data augmentation, for increasing the invariance of classifiers.




(a) MNIST (b) CIFAR-10

Figure 6.9: Invariance score versus number of additional training samples, for MNIST and CIFAR-10, with $\mathcal{T} = \mathcal{T}_{\text{sim}}$.

While random data augmentation strategies definitely improve the invariance of a classifier as we have shown above, it should be noted that better data augmentation strategies can

[4]Random transformations are constrained as follows: translation of at most 3 pixels in each direction, a scaling parameter between 0.7, and 1.3, and a rotation of at most 0.2 radians.

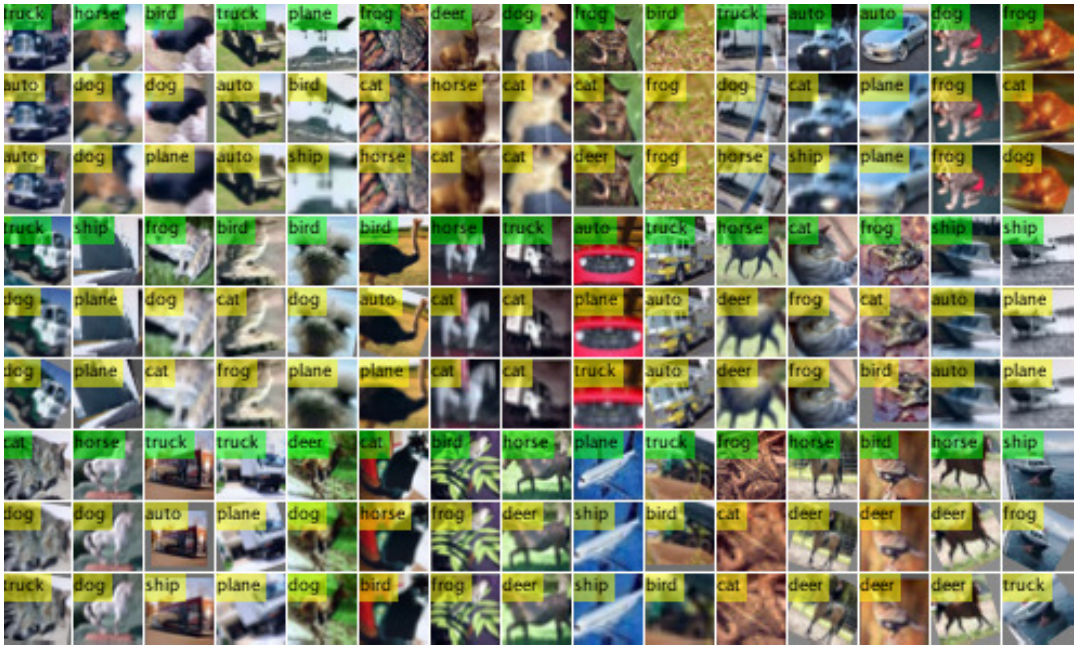


Figure 6.10: Qualitative comparison between the invariance of the original NiN network and the one trained using augmented samples, on randomly chosen samples. Images with green label represent the original images along with the correct label. The two rows below original images represent the minimally perturbed images required to modify the estimated label, respectively for the original NiN and the NiN trained with 30′000 augmented samples.

be devised for improving the invariance of classifiers [Faw+16; LCB07; Pau+14; Hau+16; CJF16]. Specifically, in [Faw+16], we propose a *worst-case* data augmentation approach, where a transformation is sought to maximize the classification loss for each image. This approach is shown to yield better invariance to geometric transformations, and better accuracy when used as a data augmentation tool, compared to the random augmentation approach. Unfortunately, this method however requires solving an optimization problem for each training point (or at least for each image belonging to a set whose size grows linearly with the number of training points), and does not therefore scale to large and high-dimensional classification tasks. A different data augmentation approach is introduced in [Pau+14], where this computational difficulty is alleviated by applying the same transformation to *all* training datapoints, and the transformation set is moreover coarsely discretized. This approach goes however against the initial intent, as different samples might require different transformations to improve the invariance of classifiers. In [Hau+16], an interesting alternative data augmentation approach is introduced, where a generative model on the transformations is first learned by registering pairs of images in the data set. Data augmentation is then performed by sampling transformations from this generative model which are then applied to the images in the training set. While yielding to a boost of the accuracy of the classifier, it should be noted that the success of this approach entirely depends on the goodness of the learned generative model. For natural images, learning such a generative model might be complex from a computational perspective in addition to the difficulty of finding registering transformations. The design of automatic and adaptive data augmentation strategies for large-scale datasets is therefore still an open problem that

deserves investigation, as the improvements on the invariance obtained with augmentation can be very large as shown above.

## 6.5 Conclusion

In this chapter, we proposed a systematic approach for measuring the invariance of any classifier to low-dimensional transformation groups. Using a manifold perspective, we were able to convert the problem of assessing the classifier's invariance to that of computing geodesic distances. We proposed a numerical algorithm based on the Fast Marching method to compute this invariance score, and applied it to several classification instances and datasets. Using the proposed approach, we also quantified the increasing invariance of CNNs with depth, and highlighted the importance of data augmentation for increasing the invariance of classifiers.

Unfortunately, the proposed approach for measuring the invariance to transformations cannot be used to model complex and potentially high-dimensional nuisance spaces such as the set of occlusions, or piecewise affine transformations. In the following chapter, we develop a probabilistic framework for studying the robustness of arbitrary classifiers to such nuisance factors.

# 7 Robustness of classifiers to complex nuisances

## 7.1 Introduction

In addition to the geometric transformations considered in the previous chapter, complex *nuisances* such as occlusions, illumination changes or image compression might affect the data in real-world classification tasks. Nuisance variables account for variability that has no effect on the result of the task, and should be ideally factored out of the classification system. The goal of this chapter is to develop a generic framework for quantitatively assessing the robustness of classifiers to such nuisance factors. While in the previous chapter, the techniques we developed were specifically crafted to low-dimensional transformation groups, the aim of this chapter is to propose a modular framework that can potentially apply to more diverse high-dimensional nuisance sets.

Specifically, we develop a general probabilistic framework for assessing and analyzing the robustness of classifiers to nuisance factors. The outcomes of the proposed framework are two-fold: the *estimation* of the robustness of classifiers to arbitrary nuisances and the *sampling* of nuisances that cause data misclassification. The latter outcome can be used to visualize and possibly improve the robustness to nuisances. In more details, we first propose a formal definition of the *average robustness to nuisance*, and provide a provably efficient Monte-Carlo estimate. In a second step, we focus on *problematic* regions of the nuisance space, where the classifier outputs low confidence scores for highly probable nuisance values. We propose a Markov Chain Monte Carlo (MCMC) sampling mechanism to quickly reach such regions of the nuisance space. This allows us to visualize problematic samples for a given classifier, and gain further insights into regions of the nuisance space which cause misclassification. Our framework is generic and can be applied to any parametrizable nuisance space and any classifier. To illustrate the proposed framework, we apply it to several classification architectures, three classification datasets and three nuisance spaces. We quantify in particular the effect of data augmentation, dropout, spatial transformer network layers [JSZ+15] on the robustness of CNNs, and compare state-of-the-art deep neural networks trained on natural image datasets in terms of their robustness to standard nuisances. Our results provide insights into the important features used by the classifier to distinguish between classes, through the visualization of the nuisances that transform an

Part of this chapter will appear in [FF16].

image to a different class. Our experiments also demonstrate that state-of-the-art classifiers are only mildly robust to standard nuisances, and that more effort should therefore be spent to improve this robustness.

This chapter is organized as follows. In Section 7.2, we introduce our probabilistic framework for assessing the robustness to nuisances of classifiers. In Section 7.3, we provide experimental results on several datasets in order to illustrate the behavior of our algorithms, and we conclude in Section 7.4.

## 7.2 Measuring the effect of nuisance variables

### 7.2.1 Definitions

We consider an arbitrary classifier that is provided through its conditional distribution $p_{\text{cl}}(c|\boldsymbol{x})$, which represents the probability that an image $\boldsymbol{x}$ is classified as $c$ by the classifier. In neural network architectures, this discrete conditional distribution $p_{\text{cl}}(\cdot|\boldsymbol{x})$ corresponds to the probability vector that can be read at the last layer of the neural network (i.e., after the softmax layer), after a feedforward pass of the input $\boldsymbol{x}$. Let $\mathcal{T}$ denote the set of nuisances, which can for example represent the set of affine transformations, diffeomorphisms, or occlusions that might corrupt the data. Similarly to the previous chapter, for a particular element in the nuisance set $\boldsymbol{\tau} \in \mathcal{T}$, we define $\boldsymbol{x}_{\boldsymbol{\tau}}$ to be the image $\boldsymbol{x}$ transformed by $\boldsymbol{\tau}$.[1] We adopt a Bayesian framework and equip the nuisance space $\mathcal{T}$ with a *prior* probability distribution $p_{\mathcal{T}}(\boldsymbol{\tau})$ that captures our region of interest in the nuisance space. For example, when $\mathcal{T}$ denotes the occlusion nuisance set, $p_{\mathcal{T}}(\boldsymbol{\tau})$ might take large values for small occlusions (covering small parts of the image), and smaller values for large occlusions. In some cases, the prior distribution $p_{\mathcal{T}}(\boldsymbol{\tau})$ might depend on the image; hence, for the sake of generality, we denote our prior distribution by $p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x})$.

We now define a quantity that allows us to measure the robustness of a classifier with respect to a nuisance set $\mathcal{T}$. Consider an image $\boldsymbol{x}$ with a ground truth label $y(\boldsymbol{x})$. The quantity $p_{\text{cl}}(y(\boldsymbol{x})|\boldsymbol{x})$ reflects the confidence that $\boldsymbol{x}$ is classified as $y(\boldsymbol{x})$, and therefore should be large when the classifier is accurate. For a given nuisance $\boldsymbol{\tau} \in \mathcal{T}$, the expression $p_{\text{cl}}(y(\boldsymbol{x})|\boldsymbol{\tau}, \boldsymbol{x}) := p_{\text{cl}}(y(\boldsymbol{x})|\boldsymbol{x}_{\boldsymbol{\tau}})$ corresponds to the probability that the transformed image $\boldsymbol{x}_{\boldsymbol{\tau}}$ is also classified as the ground truth $y(\boldsymbol{x})$. For a classifier to be robust, this quantity should also be large for typical $\boldsymbol{\tau}$. We define the robustness $\mu_{\mathcal{T}}(\boldsymbol{x})$ as the expectation of this quantity, weighted by $p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x})$:

$$\mu_{\mathcal{T}}(\boldsymbol{x}) := \int_{\boldsymbol{\tau}} p_{\text{cl}}(y(\boldsymbol{x})|\boldsymbol{\tau}, \boldsymbol{x}) p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x}) d\tau = \mathop{\mathbb{E}}_{\boldsymbol{\tau} \sim p_{\mathcal{T}}(\cdot|\boldsymbol{x})} (p_{\text{cl}}(y(\boldsymbol{x})|\boldsymbol{\tau}, \boldsymbol{x})). \tag{7.1}$$

Note that our quantity $\mu_{\mathcal{T}}(\boldsymbol{x})$ depends on the prior distribution $p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x})$; a classifier with a large $\mu_{\mathcal{T}}(\boldsymbol{x})$ will have high classification confidence in highly probable regions of the nuisance space, but $\mu_{\mathcal{T}}(\boldsymbol{x})$ is only mildly affected by the classifier confidence in low probability regions of $\mathcal{T}$. In a Bayesian inference setting, $\mu_{\mathcal{T}}(\boldsymbol{x})$ is called the *marginalized* likelihood, where the likelihood term $p_{\text{cl}}(y(\boldsymbol{x})|\boldsymbol{\tau}, \boldsymbol{x})$ is marginalized over $p_{\mathcal{T}}$.

[1] We assume in the remaining of this chapter that the nuisances $\boldsymbol{\tau}$ are represented by vectors.

Given a data distribution $p_d$, we define the *global* robustness to nuisance variables in $\mathcal{T}$ as the average of $\mu_{\mathcal{T}}(\boldsymbol{x})$ over the samples $\boldsymbol{x}$, i.e.,

$$\nu_{\mathcal{T}} := \int_{\boldsymbol{x}} \mu_{\mathcal{T}}(\boldsymbol{x}) p_{\mathrm{d}}(\boldsymbol{x}) dx = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{\mathrm{d}}} \left( \mathop{\mathbb{E}}_{\boldsymbol{\tau}\sim p_{\mathcal{T}}(\cdot|\boldsymbol{x})} [p_{\mathrm{cl}}(y(\boldsymbol{x})|\boldsymbol{\tau}, \boldsymbol{x})] \right) \tag{7.2}$$

It should be noted that the quantities $\mu_{\mathcal{T}}$ and $\nu_{\mathcal{T}}$ are bounded between 0 and 1. The global robustness $\nu_{\mathcal{T}}$ measures the average confidence that typical images perturbed with nuisances chosen according to the prior distribution $p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x})$ are classified as $y(\boldsymbol{x})$.

Observe that the approach we follow here for defining robustness in Eq. (7.1) is different from the one in Chapter 6, where the robustness to geometric transformations was measured as the distance from the identity element to the *minimal* transformation that changes the label of the classifier. Specifically, the *average* confidence computed over a user-defined prior on the nuisance space is considered in this chapter. This approach therefore does not *only* take into account the *worst-case* nuisance parameter, but all high-probability nuisance regions in the computation. This Bayesian framework, where a prior distribution on the nuisance space defines the importance of nuisance variables with respect to the classification problem, allows us to model more complex and high dimensional nuisance sets, while making the framework adaptive to the user-specific requirements that will be encoded in this prior distribution.

### 7.2.2 Estimation of the global robustness score

The global robustness measure $\nu_{\mathcal{T}}$ is a continuous quantity that involves an integration over the image space, as well as the nuisance space. We estimate these quantities using a Monte Carlo approximation method, and define the empirical quantities $\hat{\mu}_{\mathcal{T}}$ and $\hat{\nu}_{\mathcal{T}}$ as follows:

$$\hat{\mu}_{\mathcal{T}}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} p_{\mathrm{cl}}(y(\boldsymbol{x})|\boldsymbol{\tau}_i, \boldsymbol{x}), \text{ with } \boldsymbol{\tau}_i \overset{\text{iid}}{\sim} p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x}), \tag{7.3}$$

$$\hat{\nu}_{\mathcal{T}} = \frac{1}{M} \sum_{j=1}^{M} \hat{\mu}_{\mathcal{T}}(\boldsymbol{x}_j), \text{ with } \boldsymbol{x}_j \overset{\text{iid}}{\sim} p_{\mathrm{d}}. \tag{7.4}$$

$\mu_{\mathcal{T}}(\boldsymbol{x})$ is approximated by the average of the likelihood $p_{\mathrm{cl}}(y(\boldsymbol{x})|\boldsymbol{\tau}_i, \boldsymbol{x})$ over iid samples generated from the prior distribution $p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x})$. The global robustness measure is then naturally defined as the empirical average of $\hat{\mu}_{\mathcal{T}}(\boldsymbol{x}_j)$, over iid samples from the data distribution.

The computation of Eq. (7.3, 7.4) involves the transformation and classification of $NM$ samples. For computational purposes, it is therefore crucial that the empirical quantities approximate the true quantities with a small number of samples. The following result derives theoretical guarantees on the approximation error with respect to the number of samples $N$ and $M$.

**Theorem 5.** *Let $t > 0$, and $\delta \in (0, 1)$. We have $|\hat{\nu}_{\mathcal{T}} - \nu_{\mathcal{T}}| \leq t$ with probability exceeding*

$1-\delta$ *as long as*

$$M \geq \frac{\ln(2/\delta)}{2t^2}. \tag{7.5}$$

*Moreover, when the prior distributions are data-independent (i.e.,* $p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x}) = p_{\mathcal{T}}(\boldsymbol{\tau})$*), the above condition becomes*

$$NM \geq \frac{\ln(2/\delta)}{2t^2}. \tag{7.6}$$

*Proof.* Our main ingredient for proving this result is Hoeffding's inequality. We recall this inequality as follows:

**Theorem 6** (Hoeffding's inequality)**.** *Let* $(X_i, i \geq 1)$ *be a sequence of independent random variables such that* $0 \leq X_i \leq 1$*. If* $\bar{X}_n = \frac{1}{n}(X_1 + \cdots + X_n)$*, then for all* $t > 0$

$$\mathbb{P}\left(\left\{\left|\bar{X}_n - \mathbb{E}(\bar{X}_n)\right| \geq t\right\}\right) \leq 2\exp\left(-2nt^2\right).$$

**Case (a).** We start our proof by considering the case where the prior distribution does not depend on the image: $p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x}) = p_{\mathcal{T}}(\boldsymbol{\tau})$, to establish the result in Eq. (7.6). We have:

$$\nu_{\mathcal{T}} = \int_{\boldsymbol{x}}\int_{\boldsymbol{\tau}} p_{\mathrm{cl}}(y(\boldsymbol{x})|\boldsymbol{x},\boldsymbol{\tau}) p_{\mathcal{T}}(\boldsymbol{\tau}) p_d(\boldsymbol{x}) d\boldsymbol{\tau} d\boldsymbol{x},$$
$$\hat{\nu}_{\mathcal{T}} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} p_{\mathrm{cl}}(y(\boldsymbol{x}_j)|\boldsymbol{x}_j,\boldsymbol{\tau}_i) := \frac{1}{M}\frac{1}{N}\sum_{j=1}^{M}\sum_{i=1}^{N} Z_{j,i}.$$

The random variables $\boldsymbol{\tau}_i$ and $\boldsymbol{x}_j$ are independent, hence $\{Z_{j,i}\}_{(j,i)}$ are pairwise independent. Note moreover that $Z_{j,i} \in [0,1]$, and that $\mathbb{E}(Z_{j,i}) = \nu_{\mathcal{T}}$ for any $j, i$. Hence, by applying Hoeffding's inequality, we obtain

$$\mathbb{P}\left(|\hat{\nu}_{\mathcal{T}} - \nu_{\mathcal{T}}| \geq t\right) \leq 2\exp(-2NMt^2).$$

Setting $\delta = 2\exp(-2NMt^2)$, we obtain the desired result in Eq.(7.6).

**Case (b).** We now consider the general case where the the prior distribution $p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x})$ *depends* on the image, and our goal is to establish the result in Eq. (7.5). We have:

$$\nu_{\mathcal{T}} = \int_{\boldsymbol{x}}\int_{\boldsymbol{\tau}} p_{\mathrm{cl}}(y(\boldsymbol{x})|\boldsymbol{x},\boldsymbol{\tau}) p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x}) p_d(\boldsymbol{x}) d\boldsymbol{\tau} d\boldsymbol{x},$$
$$\hat{\nu}_{\mathcal{T}} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} p_{\mathrm{cl}}(y(\boldsymbol{x}_j)|\boldsymbol{x}_j,\boldsymbol{\tau}_i) := \frac{1}{M}\frac{1}{N}\sum_{j=1}^{M}\sum_{i=1}^{N} Z_{j,i}.$$

In this case, the random variables $Z_{j,i}$ and $Z_{j,i'}$ are not independent in general (for $i \neq i'$).

We therefore introduce the random variable

$$W_j = \frac{1}{N}\sum_{i=1}^{N} Z_{j,i},$$

and note that $\{W_j\}_j$ are pairwise independent, as the random variables $\{\boldsymbol{x}_j\}$ are chosen independently. Note moreover that $\mathbb{E}(W_j) = \mathbb{E}(Z_{j,i}) = \nu_{\mathcal{T}}$, and that $W_j \in [0,1]$. We apply Hoeffding's inequality for $W_j$ and obtain

$$\mathbb{P}\left(|\hat{\nu}_{\mathcal{T}} - \nu_{\mathcal{T}}| \geq t\right) \leq 2\exp(-2Mt^2).$$

By setting $\delta = 2\exp(-2Mt^2)$, we obtain the desired result in Eq.(7.5). □

For prior distributions on nuisance spaces that are independent of the datapoint $\boldsymbol{x}$, the above result shows that, by choosing $N$ and $M$ in the order of 100, one can obtain very accurate estimates for $\hat{\nu}_{\mathcal{T}}$. When the nuisance prior is data-dependent, the worst-case result becomes independent of $N$, and one needs more samples to derive accurate estimates. In many cases of interest however, the independent case applies as the prior distribution does not significantly differ for different images. It should finally be noted that the bounds in Theorem 5 do not depend on the dimension of the nuisance space; this shows that the approximate quantity $\hat{\nu}_{\mathcal{T}}$ can be very accurate (for moderately large $N$ and $M$) even for high dimensional nuisance spaces.

### 7.2.3 Estimation of the problematic nuisances

While $\nu_{\mathcal{T}}$ measures the *average* likelihood of the classifier (i.e., confidence of correct classification, when nuisance samples are drawn from the prior distribution), it is also crucial to visualize and understand the *problematic* regions of the nuisance space where the classifier has low confidence on transformed images. The problematic regions of the nuisance space are mathematically described by the *posterior* distribution $p_{\text{cl}}(\boldsymbol{\tau}|\overline{y(\boldsymbol{x})}, \boldsymbol{x})$, where we define $p_{\text{cl}}(\overline{y(\boldsymbol{x})}|\boldsymbol{\tau}, \boldsymbol{x}) = 1 - p_{\text{cl}}(y(\boldsymbol{x})|\boldsymbol{\tau}, \boldsymbol{x})$ to be the probability that $\boldsymbol{x}_{\boldsymbol{\tau}}$ is *not* classified as $y(\boldsymbol{x})$. Sampling from this posterior distribution allows us to "diagnose" the set of nuisance parameters that can cause classification errors. Using the Bayes rule, the posterior distribution can be written as the normalized product of the likelihood and prior distribution

$$p_{\text{cl}}(\boldsymbol{\tau}|\overline{y(\boldsymbol{x})}, \boldsymbol{x}) = \frac{1}{Z} p_{\text{cl}}(\overline{y(\boldsymbol{x})}|\boldsymbol{\tau}, \boldsymbol{x}) p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x}),$$

with $Z$ the normalizing constant. It should be noted that this posterior distribution is typically a complex high dimensional distribution, where specialized sampling algorithms do not apply. To sample from this posterior distribution, we adopt here the celebrated Metropolis MCMC method for sampling from high dimensional distributions [Met+53]. The sample values are produced iteratively, where the distribution of the next sample depends only on the current sample value (hence making the samples sequence a Markov Chain). At each iteration, the algorithm picks a candidate for the next sample by sampling from a *proposal distribution* $q$, which guides the exploration of the nuisance space $\mathcal{T}$. Then, with some probability $p_{\text{accept}}$, the candidate is either accepted, in which case the candidate value

**Algorithm 5** Metropolis algorithm for sampling from $p_{\text{cl}}(\boldsymbol{\tau}|\overline{y(\boldsymbol{x})}, \boldsymbol{x})$

**Initialization:** Start with a randomly initialized sample in the nuisance set $\boldsymbol{\tau}^{(0)} \in \mathcal{T}$.
**For each iteration $s$ of the random walk on the nuisance space $\mathcal{T}$, do:**
Draw a sample $\boldsymbol{\tau}' \sim q(\boldsymbol{\tau}|\boldsymbol{\tau}^{(s)})$.
Let

$$p_{\text{accept}} = \min\left(1, \frac{p_{\text{cl}}(\overline{y(\boldsymbol{x})}|\boldsymbol{\tau}', \boldsymbol{x}) p_{\mathcal{T}}(\boldsymbol{\tau}'|\boldsymbol{x}) q(\boldsymbol{\tau}^{(s)}|\boldsymbol{\tau}')}{p_{\text{cl}}(\overline{y(\boldsymbol{x})}|\boldsymbol{\tau}^{(s)}, \boldsymbol{x}) p_{\mathcal{T}}(\boldsymbol{\tau}^{(s)}|\boldsymbol{x}) q(\boldsymbol{\tau}'|\boldsymbol{\tau}^{(s)})}\right).$$

Generate a uniform sample in $u \in [0, 1]$.
If $u \leq p_{\text{accept}}$, $\boldsymbol{\tau}^{(s+1)} \leftarrow \boldsymbol{\tau}'$; otherwise, $\boldsymbol{\tau}^{(s+1)} \leftarrow \boldsymbol{\tau}^{(s)}$.

is used in the next iteration, or rejected. The acceptance probability is controlled by the ratio between the probability of the posterior distribution at the candidate sample to that of the current sample. The algorithm is summarized in Algorithm 5.

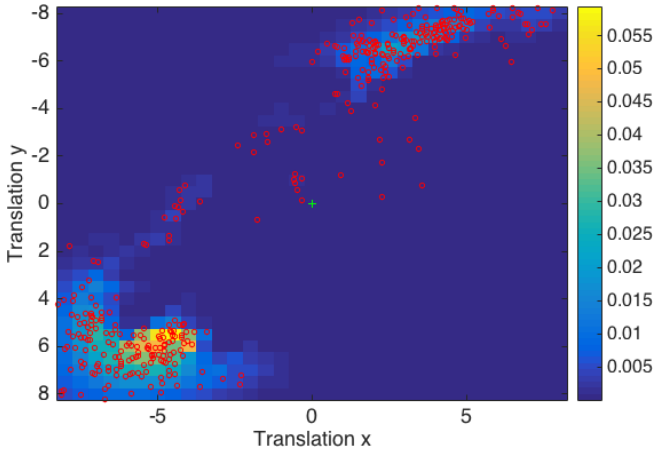



Figure 7.1: Example map of the (un-normalized) posterior distribution $p_{\text{cl}}(\boldsymbol{\tau}|\overline{y(\boldsymbol{x})}, \boldsymbol{x})$ when $\mathcal{T} =$ 2d translations. We overlay samples obtained using the Metropolis MCMC method.

In practice, we set the proposal distribution $q(\cdot|\boldsymbol{\tau}) \sim \mathcal{N}(\boldsymbol{\tau}, \sigma^2_{\text{prop}}\boldsymbol{I})$. It should be noted that the above algorithm can be applied to any parametrizable nuisance space $\mathcal{T}$, and any prior distribution $p_{\mathcal{T}}$ (in particular prior distributions where sampling is difficult, as sampling from this prior distribution is not required) in order to find problematic samples of the nuisance space. Fig. 7.1 illustrates the samples drawn using the Metropolis algorithm when $\mathcal{T}$ is the set of 2D translations of an arbitrary image and baseline classifier. It can be seen that samples obtained with Metropolis confine to highly probable regions of the nuisance space (these correspond to nuisance parameters with *low* classification confidence). In particular, it should be noted that the Metropolis method relying on a Markov Chain random walk for sampling is much more efficient than the standard approach where independent samples are drawn from $p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x})$, and accepted or rejected depending on the values of their likelihood. This Metropolis method is therefore particularly suited to our framework, as it can efficiently find "problematic samples", even if the average score $\mu_{\mathcal{T}}(\boldsymbol{x}) \approx 1$.

## 7.3 Experimental evaluation

### 7.3.1 MNIST handwritten digits

We evaluate in this section different classifiers in terms of their robustness to the set $\mathcal{T}$ of affine transformations. We parametrize the elements in $\mathcal{T}$ with vectors $\boldsymbol{\tau} \in \mathbb{R}^6$ representing the column-reshaped $2 \times 3$ standard matrix representations of affine transformations. We consider a Gaussian prior distribution on $\mathcal{T}$ given by $p_{\mathcal{T}} = \mathcal{N}(\mathbf{e}, \boldsymbol{\Sigma})$, where $\mathbf{e}$ is the identity affine transformation, and $\boldsymbol{\Sigma} \in \mathbb{R}^{6\times 6}$ is a covariance matrix that penalizes large changes in the appearance of the image. To define the notion of *appearance change*, we follow a similar approach to that used in the previous chapter. Specifically, we quantify the change in appearance between two elements $\boldsymbol{\tau}_0$ and $\boldsymbol{\tau}_1$ in $\mathcal{T}$ using the geodesic distance on the manifold of transformed samples $\{\boldsymbol{x}_{\boldsymbol{\tau}} : \boldsymbol{\tau} \in \mathcal{T}\}$. Recall that this distance can be written

$$d(\boldsymbol{\tau}_0, \boldsymbol{\tau}_1) = \inf_{\gamma} \int_0^1 \sqrt{\boldsymbol{\gamma}(t)^T \boldsymbol{G}_{\boldsymbol{\gamma}(t)} \boldsymbol{\gamma}(t)} dt, \tag{7.7}$$

where the infimum is taken over all $C^1$ curves $\boldsymbol{\gamma}$ that satisfy $\boldsymbol{\gamma}(0) = \boldsymbol{\tau}_0$ and $\boldsymbol{\gamma}(1) = \boldsymbol{\tau}(1)$, and $\boldsymbol{G}$ denotes a Riemannian metric on the manifold $\mathcal{T}$. When $\tau_1$ is in the neighborhood of $\boldsymbol{\tau}_0$, we can approximate the matrix $\boldsymbol{G}_{\boldsymbol{\gamma}(t)}$ (for any $t$) by $\boldsymbol{G}_{\boldsymbol{\tau}_0}$, provided $\boldsymbol{G}_{\boldsymbol{\gamma}(\boldsymbol{t})}$ is slowly varying with $\boldsymbol{\gamma}(t)$. By assuming a constant $\boldsymbol{G}_{\boldsymbol{\gamma}(t)} = \boldsymbol{G}_{\boldsymbol{\tau}_0} = \boldsymbol{G}$, the distance in Eq. (7.7) can be computed in closed-form:

$$d(\boldsymbol{\tau}_0, \boldsymbol{\tau}_1) = \sqrt{(\boldsymbol{\tau}_1 - \boldsymbol{\tau}_0)^T \boldsymbol{G} (\boldsymbol{\tau}_1 - \boldsymbol{\tau}_0)}.$$

We therefore naturally set the prior distribution on $\mathcal{T}$ in order to penalize large variations in the appearance of the image, by defining

$$p_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{x}) \propto \exp\left(-\alpha d(\mathbf{e}, \boldsymbol{\tau})^2\right) = \exp\left(-(\boldsymbol{\tau} - \mathbf{e})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\tau} - \mathbf{e})\right),$$

with $\boldsymbol{\Sigma}^{-1} = \alpha \boldsymbol{G}$, and $\alpha$ is a parameter controlling the "magnitude" of the transformation. In that sense, our prior distribution penalizes changes in the *appearance* of the image (assuming a constant Riemannian metric), and favors nuisance regions that do not significantly distort the data. While we use this prior distribution in this experiment, it should be noted that our framework is generic, and not limited to such a prior. We show in Fig. 7.3 transformed versions of arbitrary MNIST images with transformations drawn from the prior distribution using different values of $\alpha$.

We consider two baseline CNN architectures on the MNIST task, CNN-1 and CNN-2, of respectively 1 and 2 hidden layers. We then consider the following modifications of these baseline neural networks:

- **Dropout regularization**: We use a dropout regularization (with probability $p = 0.5$) at the last fully connected layer of the network,
- **Data Augmentation (DA)**: At the training stage, we apply a small random translation to the samples with probability 0.1. In other words, we randomly translate 10% of the samples at the training stage.
- **Spatial Transformer Network (STN)** [JSZ+15]: We use a model where the

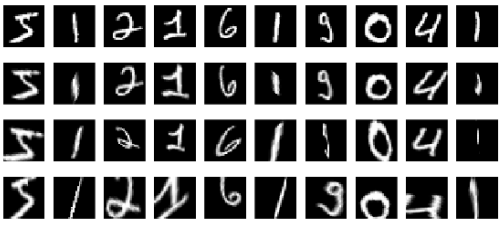


Figure 7.2: Original images are shown in row 1. Samples drawn from prior distribution with $\alpha = 100$ [row 2, mild transformations], $\alpha = 50$ [row 3, medium transformations], and $\alpha = 10$ [row 4, severe transformations].

| Model | Test error (%) | $\hat{\nu}_{\mathcal{T}}$ ($\alpha = 100$) | $\hat{\nu}_{\mathcal{T}}$ ($\alpha = 50$) | $\hat{\nu}_{\mathcal{T}}$ ($\alpha = 10$) |
|---|---|---|---|---|
| CNN-1 | 1.26 | 0.90 | 0.76 | 0.30 |
| + **Dropout** | 0.88 | 0.90 | 0.77 | 0.31 |
| + **DA** | 1.04 | 0.93 | 0.85 | 0.44 |
| + **STN** | 0.93 | **0.96** | **0.90** | 0.52 |
| CNN-2 | 1.16 | 0.94 | 0.83 | 0.36 |
| + **Dropout** | **0.68** | 0.93 | 0.82 | 0.37 |
| + **DA** | 1.09 | 0.94 | 0.87 | 0.48 |
| + **STN** | 0.79 | **0.96** | **0.90** | **0.53** |

Table 7.1: Robustness to affine transformations of several networks on the MNIST dataset. Each network is trained for 50 epochs.

> localization network is a two layer CNN which operates on the image input. The output from the localization network is a 6 dimensional vector specifying the parameters of the affine transformation. This network is trained with data augmentation.

Dropout, DA and STN are often used in order to improve the classification performance. The goal here is to see the effect of these techniques on the robustness to nuisance factors.

Table 7.1 reports the affine robustness $\hat{\nu}_{\mathcal{T}}$ with $N = M = 1000$ for the different networks for three transformation regimes (mild, medium and severe transformations respectively obtained by setting $\alpha = 100, 50, 10$). By comparing CNN-1 and CNN-2, it can be seen that increasing the number of layers leads to a better affine invariance of the model. This result is in line with the conclusions of the previous chapter showing that an increase in the number of layers of a deep convolutional network leads to improved robustness to similarity transformations. While dropout regularization leads to significant improvement in test accuracy, it has barely any effect on the robustness of the classifier to affine transformations. This shows that robustness and test accuracy capture two different properties of the classifier. In fact, while the robustness property measures the effect of nuisance variables that might occur in real-world applications on the classification function, the test set usually contains a restricted set of images following the same distribution as the training set. Conversely, data augmentation (with translated samples) has led in this example to a decrease in the test accuracy, while boosting the robustness to transformations. Moreover, the addition of STN layers also improves the robustness of classifiers to transformations in the data.

Among the tested classifiers, CNN-2-STN has the maximum robustness for all parameters $\alpha$,

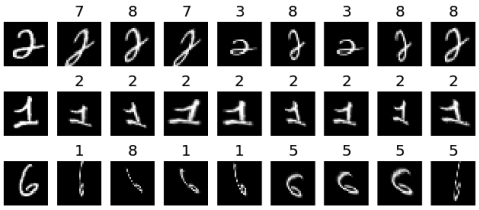



Figure 7.3: Samples drawn from the posterior distribution $p(\boldsymbol{\tau}|\overline{y(\boldsymbol{x})}, \boldsymbol{x})$ with $\alpha = 100$. On the left, the original image, and then the transformed images with nuisances sampled from the posterior distribution for the CNN-2 with Spatial Transformer Network. The estimated label by the classifier of each transformed image is shown on top of each image. All shown images are misclassified by the classifier.

with an robustness score larger than 0.9 for mild and medium transformations ($\alpha = 100, 50$). In other words, the classifier correctly classifies transformed samples with confidence surpassing 90%. Nevertheless, despite these large *average* scores, this same network can wrongly classify images that are however quite easily identifiable by a human observer. To see this, we show in Fig. 7.3 transformed images with samples drawn from the posterior distribution $p_{\text{cl}}(\boldsymbol{\tau}|\overline{y(\boldsymbol{x})}, \boldsymbol{x})$ using the sampling technique of Section 7.2.3.[2] Quite interestingly, these samples have a large variation, thereby showing that multiple regions of the nuisance space of the classifier can cause data misclassification. For example, relatively small transformations of a digit 2 can cause it to be a 7, 8 or 3. This shows the existence of many "directions" that potentially cause the classifier to misclassify.

### 7.3.2 Natural images classification

| VGG-CNN-S | VGG-16 | VGG-19 | GoogLeNet |
|---|---|---|---|
| 0.62 | 0.68 | 0.68 | 0.67 |

Table 7.2: Robustness to piecewise affine transformations $\hat{\nu}_{\mathcal{T}}$ of different networks trained on ImageNet

We now conduct experiments on deep neural networks that are trained on the ImageNet dataset. Specifically, we consider 4 different pre-trained networks: VGG-CNN-S [Cha+14], VGG-16, VGG-19 [SZ14], and GoogLeNet [Sze+15]. We evaluate the robustness of these networks to *piecewise* affine transformations. Specifically, the image is divided into cells, and each cell undergoes a different affine transformation. We parametrize the transformations using motion vectors defined for regularly spaced control points in the image. More precisely, a transformation is parametrized by a set of motion vectors stacked in an array $\mathbf{V} \in \mathbb{R}^{2\times L}$, where $L$ defines the number of control points. We then define a prior distribution $p_{\mathcal{T}} = \mathcal{N}(\mathbf{0}_{2L}, \boldsymbol{\Sigma})$, where $\mathbf{0}_{2L}$ denotes the zero motion vector, and where $\boldsymbol{\Sigma}$ is a covariance matrix whose correlations decay with the distance between control points. Specifically, if $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$ denote two control points, we set the correlation between these points to be $\frac{\sigma^2}{\|\mathbf{u}-\mathbf{v}\|_2^2}$, where $\sigma$ is a constant controling the magnitude of the correlations. In the experiments below, we use $\sigma = 20$. This prior distribution, which forces nearby control points to have similar motion vectors, incorporates a smoothness constraint on the set of

[2] We post-processed the samples obtained using Metropolis (section 7.2.3) by keeping only the samples having a label different than $y(\boldsymbol{x})$. The depicted samples are randomly chosen from this set.

transformations, and results in having well-behaved and natural transformations. It should be noted that the work in [Fre+15] defined a similar prior distribution on motion fields in a different context. We illustrate in Fig. 7.4 images transformed with nuisance parameters sampled from the introduced prior distribution.

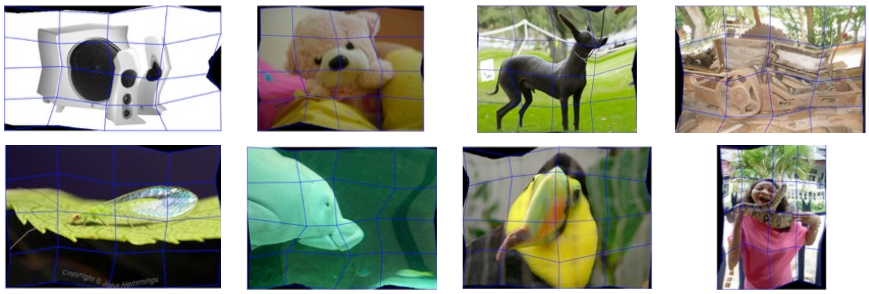


Figure 7.4: Transformed versions of images taken from the ILSVRC 2012 validation dataset.

We report the robustness measures $\hat{\nu}_{\mathcal{T}}$ of the different networks in Table 7.2, for $M = 200$ and $N = 100$. It can be noted that VGG-CNN-S is slightly worse than other networks in terms of robustness to piecewise affine transformations. We recall that numbers in Table 7.2 indicate the confidence of the classifier on transformed samples; for example, the VGG-CNN-S correctly classifies transformed samples (see examples of images in 7.4) with confidence of 62%. The comparison between the different classifiers confirms once again the result highlighted in the previous section, namely that depth improves the robustness to nuisance factors (in particular piecewise affine transformations), as VGG-16, 19 and GoogLeNet contain substantially more layers than VGG-CNN-S. The overall scores shown in Table 7.2 show however that these state-of-the-art networks correctly classify samples with confidence lower than 70%, for small enough piecewise affine transformations of the data (see Fig. 7.4 for example images).

We visualize images with nuisances sampled using the proposed method in 7.2.3 from the posterior $p_{\text{cl}}(\boldsymbol{\tau}|\overline{y(\boldsymbol{x})}, \boldsymbol{x})$ in Fig. 7.5 for the different networks. For some examples, a "natural" transformation of the image leads to a label change: observe that the "Gyromitra" is indeed transformed to be visually similar to an image representing a "hen". These examples provide insights into *the concepts* that the deep network uses to discriminate between the classes. In particular, observe that the required nuisance parameter $\boldsymbol{\tau}$ to transform a "white wolf" onto an "arctic fox" or "Samoyed" is rather intuitive for a human. In particular, the deep network heavily relies on the deformation of the "nose" cue in order to change the estimated label, and therefore uses this cue in order to distinguish between these neighbouring classes. It should be noted however that in other cases, the transformation is not well interpretable from a human perspective. Specifically, in many images, relatively small transformations are sufficient to change the image class to labels that are very different from a human perspective (e.g., lampshade $\rightarrow$ sea slug, necklace, ...). This shows deficiencies in the concepts learned by these classifiers, and that the context of the image is probably not sufficiently used to infer the label (e.g., the context of a scene representing a lamp shade is very different from sea slug). In fact, while it is possible to modify the geometric aspect of an object (e.g., lampshade) using a nuisance transformation from $\mathcal{T}$, the overall scene

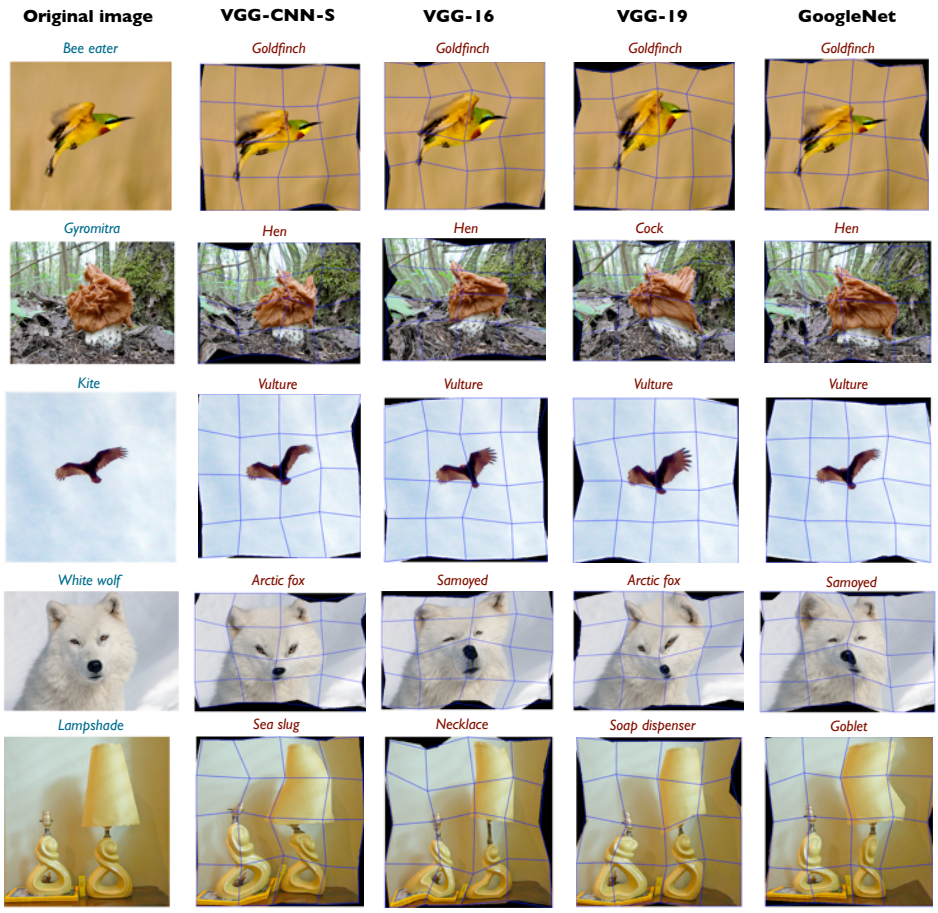
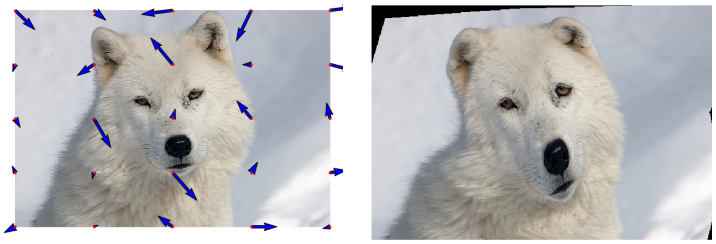



Figure 7.5: Robustness of different networks trained on ImageNet to piecewise affine transformations. The left column displays original images, and the other columns show the transformed images, where transformations are sampled from the posterior $p_{\text{cl}}(\boldsymbol{\tau}|\overline{y(\boldsymbol{x})}, \boldsymbol{x})$ for 4 different classifiers. The estimated label of each image is shown on top. A post-processing step was applied similarly to the experiment in Fig. 7.3 (see footnote 2).

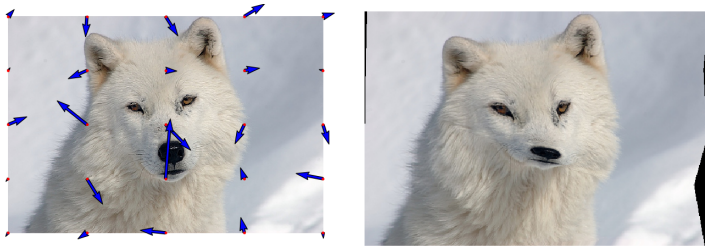
context (characterized by the shadings, neighbouring objects, etc...) is much more difficult to alter and should ideally be properly modeled by the classifier to achieve robustness.

To further understand the features learned by the classifier to discriminate between two specific classes, we apply the proposed sampling mechanism in Section 7.2.3, but this time with a slightly modified likelihood function. Specifically, given a *target label* $t \neq y(\boldsymbol{x})$, we set the likelihood to be the probability that the transformed image, $\boldsymbol{x}_{\boldsymbol{\tau}}$, is classified as $t$. Formally, the likelihood is given by $p_{\text{cl}}(t|\boldsymbol{\tau}, \boldsymbol{x})$. This view slightly differs from the formulation in Section 7.2.3 and experiment in Fig. 7.5, where the likelihood represented the probability of the transformed image to be classified as *any* other class (different than the original class, $y(\boldsymbol{x})$); i.e., $p_{\text{cl}}(\overline{y(\boldsymbol{x})}|\boldsymbol{\tau}, \boldsymbol{x})$. Instead, we now *fix* the target label $t$, and wish to sample nuisance variables such that the transformed image is confidently labeled as $t$. By doing so, we visualize the differences between two classes from the point of view of the deep neural network. In our experiment, we considered a fixed original image $\boldsymbol{x}$ (with label "white wolf"), and we ran the experiment three times, each time with a different target label ("polar bear", "arctic fox", and "Samoyed"). We illustrate in Fig. 7.6 the *average* transformations

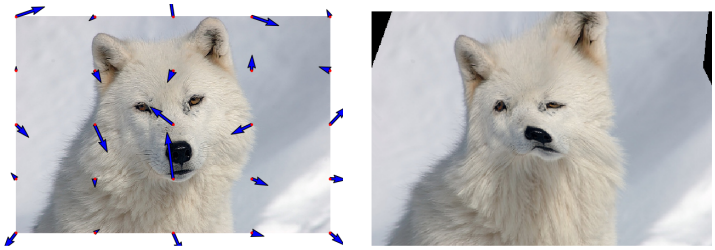
obtained when sampling from the posterior distribution $p_{\text{cl}}(\boldsymbol{\tau}|t, \boldsymbol{x})$, for the three different experiments.[3] The depicted results show that the deep neural network captures, in this case, intuitive and interpretable features to distinguish between the different animals. Specifically, the transformation from a "white wolf" to a "polar bear" involves enlarging the nose, and reducing the size of the ears. Moreover, the transformation to an "arctic fox" transforms the nose. Interestingly (and perhaps surprisingly), the network succeeds in finding very plausible transformations (from the human point of view) for this example, which shows that the network uses the correct features in these binary classification tasks. It should nevertheless be noted that, for other images, the network can change the estimated label with a small transformation that is difficult to interpret (see Fig. 7.5).



(a) Polar bear



(b) Arctic fox



(c) Samoyed

Figure 7.6: How to transform a white wolf into (a) a polar bear, (b) an Arctic fox, (c) a Samoyed dog? For each of the three target labels, the left image represents the motion vectors of the *average* sampled transformations. For clarity, we overlayed on top of the motion vectors the original image classified as "white wolf". The right image depicts the result of applying this (average) transformation to the original "white wolf" image. Experiments performed on the VGG-16 classifier.

[3]Similarly to the previous experiments, we restrict ourselves to the images that are misclassified. The average is therefore taken over the transformations that have an estimated label $t$.

### 7.3.3 Face recognition

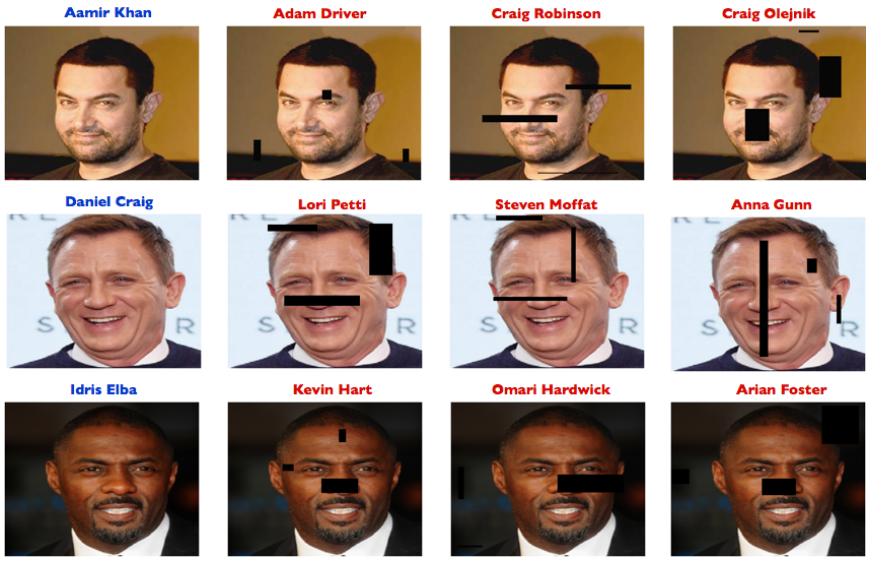



Figure 7.7: Robustness of VGG-Faces classifier to artificial occlusion. Left column: original image, with correct label. Columns 2 to 4 are samples from the posterior distribution. On top of each image, we indicate the *estimated* label. A post-processing step was applied similarly to the experiment in Fig. 7.3 (see footnote 2).

We finally consider a face recognition application, where we consider the very recent VGG-Face classifier from [PVZ15], and measure the robustness of this classifier to simple *occlusions*. Specifically, we consider a nuisance set $\mathcal{T}$ where $b$ occluding rectangles corrupt the images: any pixel belonging to one of the rectangles is "erased" and set to zero. We consider a prior probability distribution on this nuisance space that penalizes the total area of occluded pixels to favor small occluding boxes. Specifically, we set

$$p_{\mathcal{T}}(\boldsymbol{\tau}) \propto \exp\left(-O_p/\sigma^2\right), \tag{7.8}$$

where $O_p$ is the number of occluded pixels by the $b$ rectangles, and $\boldsymbol{\tau} \in \mathbb{R}^{4b}$ is a parametrization of the state consisting of $b$ rectangles, each parametrized with 4 scalars (upper left and lower right point). In the experiments, we set $\sigma = 2000$, $b = 3$. For the Metropolis algorithm, we use a Gaussian proposal with standard deviation $\sigma_{\text{prop}} = 5$, and set the number of iterations to 1000. To favor diverse samples from the posterior, we perform several runs with different random initializations.

We illustrate different samples from the posterior distribution $p_{\text{cl}}(\boldsymbol{\tau}|\overline{y(\boldsymbol{x})}, \boldsymbol{x})$ in Fig. 7.7. Interestingly, it can be seen that with relatively small occluding boxes, one can change the estimated label of the classifier. More surprising, these simple occlusions can cause *trivial* errors in the estimated label (e.g., *Aamir Khan* $\rightarrow$ *Craig Robinson*, or *Daniel Craig* $\rightarrow$ *Anna Gunn*). This lack of robustness is specifically problematic in a face recognition system as it can be exploited by intruders for fraudulous identification in systems using face recognition. The proposed sampling tool is thus important to assess the robustness to such nuisances, and to reveal the weaknesses of such classifiers before their deployment in possibly hostile environments. It should be noted moreover that the proposed sampling approach explores

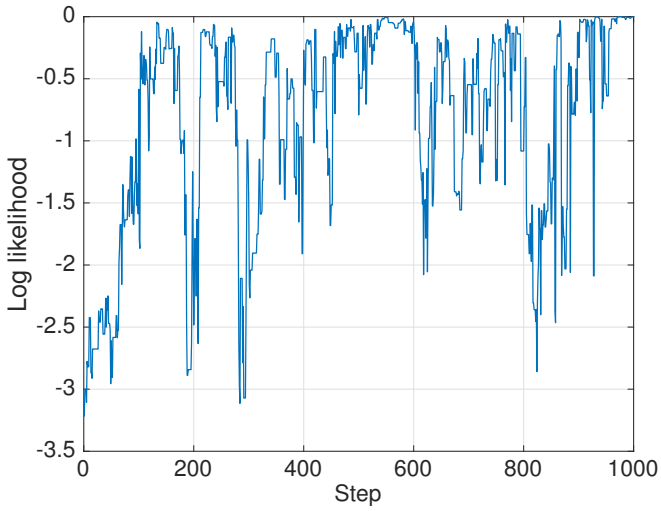



Figure 7.8: Evolution of the log-likelihood $\log(p_{\text{cl}}(\overline{y(\boldsymbol{x})}|\boldsymbol{\tau}, \boldsymbol{x}))$ in one run of the Metropolis algorithm.

diverse nuisance parameters causing data misclassification, and does not seek to obtain the *minimal* nuisance parameter causing misclassification. We illustrate in Fig. 7.8 the evolution of the log-likelihood term (i.e., the log of the misclassification probability $\log(p_{\text{cl}}(\overline{y(\boldsymbol{x})}|\boldsymbol{\tau}, \boldsymbol{x}))$) in the Metropolis algorithm. Notice that, after approximately 100 iterations, the algorithm reaches regions of the nuisance space with large likelihood. Note also that the log-likelihood is *not* monotonously increasing, which suggests that the algorithm is not stuck in a single region of the nuisance space. It should further be noted that, similarly to the visualization in Fig. 7.5, one can draw conclusions on the features used in the face recognition. Specifically, we observed that in many cases, the classifier changes label by adding a relatively small occluding box on the person's nose (and not the eyes, as one would expect), which shows that this represents an important feature in this automatic face recognition system. To illustrate this point, we show in Fig. 7.9 the *average* samples from the posterior causing data misclassification. It can be seen that the occluding boxes largely concentrate around the nose, which shows the importance of this cue for this classifier.

## 7.4 Conclusion

We proposed a simple and generic probabilistic framework for measuring the average robustness to nuisance variables, as well as for sampling problematic nuisance variables. Our framework can deal with any type of parametrizable nuisance factors, as long as a prior distribution that defines the region of interest on this space is defined. The proposed tool allows us to discover the "weak spots" of any given classifier by appropriately sampling likely nuisance vectors that cause misclassification. Moreover, the visualization of problematic samples provides insights onto the features learned by the system. We believe that the proposed framework is not only an important tool to assess the robustness of a classifier under unexpected nuisance variations, but that it will also open new possibilities for improving the robustness of classifiers to specific nuisances.

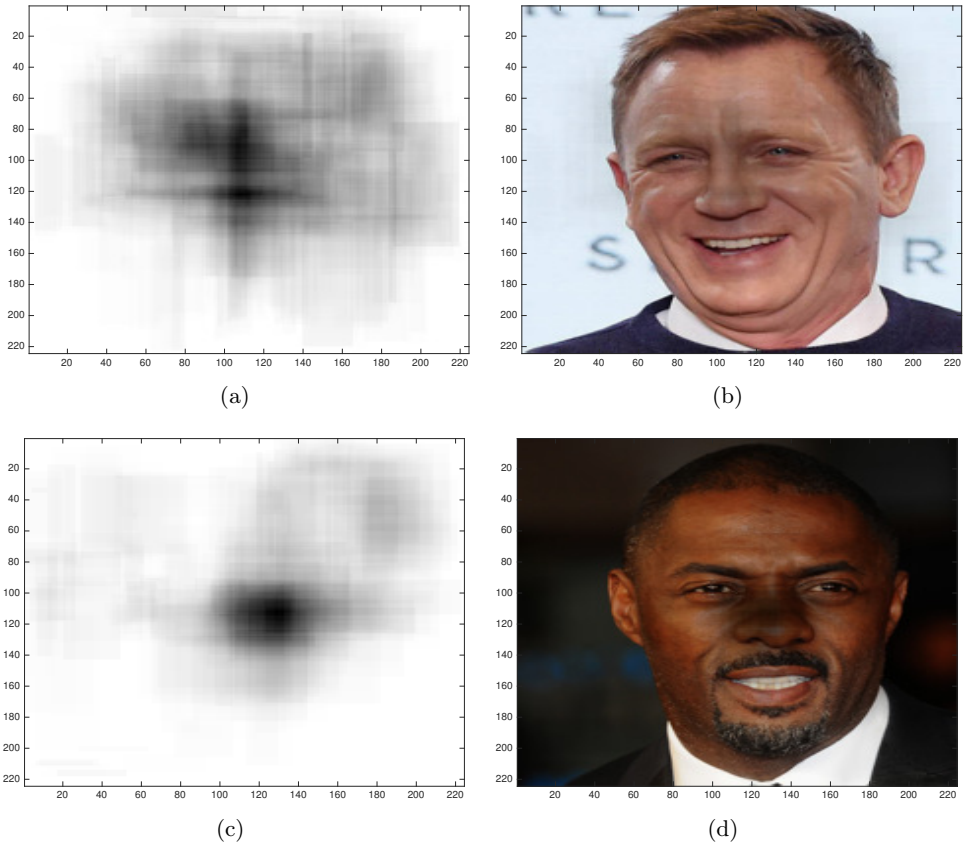



Figure 7.9: Average over all nuisance samples from the posterior leading to a misclassification, for two different images. Left: the nuisance parameters are illustrated without the original image. Right: same image, where the face image is shown in the background.

# 8 Conclusions

## 8.1 Summary

In this thesis, we analyzed the robustness of classifiers to a wide range of perturbations in the data, spanning from adversarial noise to random noise, as well as more structured nuisances such as geometric transformations and occlusions.

We first studied the *adversarial* robustness of classifiers, where data points undergo *minimal* perturbations that change the estimated label of the classifier. The computation of adversarial perturbations specifically requires solving a non-convex optimization problem that uses the knowledge of the classification model. Our first contribution in this thesis was a new optimization algorithm to efficiently estimate adversarial perturbations. Through extensive simulations, we showed that the proposed optimization algorithm outperforms existing methods in the task of robustness estimation, and can lead to an improvement in the robustness when fine-tuning with perturbed data points. Our study moreover provided a confirmation that current classifiers are extremely unstable to adversarial perturbations, despite achieving impressive classification accuracy. Next, we proposed a theoretical analysis of the instability of classifiers to adversarial perturbations, and showed the existence of learning-independent limits on the robustness of classifiers. These limits are derived for fixed classification families (e.g., the family of linear classifiers), and show that in common classification tasks, it is not possible to find robust *and* accurate classifiers in the family independently of the training procedure. We specifically quantified this tradeoff for linear and quadratic classifiers, and showed that the fundamental limit on the robustness increases with the flexibility of the classification family.

Then, we analyzed quantitatively the robustness of classifiers in a novel noise regime that unifies adversarial and random noise. We specifically derived upper and lower bounds on the robustness of classifiers in this generalized noise regime based on the curvature of the classifier's decision boundary. Provided the curvature is sufficiently small, our bounds show specifically that robustness to random noise can be achieved in high dimensional classification tasks, even if the classifier has a poor robustness to adversarial perturbations. This result quantitatively justifies empirical observations showing that state-of-the-art classifiers are robust to random noise, even when the same classifiers have very low robustness to adversarial perturbations. Our bounds also show that these classifiers remain unstable to *semi*-random noise that is mostly random, and only slightly adversarial.

We finally presented methods for quantifying the robustness of classifiers to structured perturbations encountered in computer vision classification tasks, such as geometric transformations and occlusions. Specifically, we first proposed a method for quantifying invariance to geometric transformations. Our novel invariance score is based on the distance between the identity transformation and the minimal transformation required to change the estimated label of the classifier. To precisely define this distance on the transformation space, we used a manifold representation of the set of transformed images, and defined the distance to be the geodesic distance on the manifold. We proposed a numerical algorithm for the computation of this invariance measure, and showed that state-of-the-art classifiers are not robust to small similarity transformations. Using our invariance score, we highlighted moreover that adding transformed samples to the training set can lead to a significant increase in the invariance score. We finally extended this idea of quantifying the robustness to general nuisance spaces. To do so, we specifically proposed a novel probabilistic model, where the nuisance space was equipped with a prior distribution that captures the region of interest, and where the likelihood associated to a nuisance parameter is defined as the probability of misclassification of the transformed image. Using this model, we provided methods for estimating the robustness, and for sampling problematic nuisance vectors from the nuisance set. The visualization of problematic samples allows us to explore the different regions of the nuisance space causing misclassification, and further provides insights onto the discriminative features used by the classifier.

In summary, this thesis offers important theoretical and empirical results in the analysis of the robustness of classifiers to a diverse set of perturbations. By identifying and thoroughly analyzing the quantities that affect the robustness of classifiers, our results moreover pave the way towards designing classifiers with improved robustness properties.

## 8.2 Future directions

Several questions related to the adversarial instability of classifiers remain open. In particular, while we thoroughly studied fundamental limits on the adversarial robustness for the family of linear and quadratic classifiers, establishing such limits for more complex classifiers deserves more investigation. We believe that it might be possible to derive similar bounds for more complex classifiers by using the general result in Lemma 1, along with explicit parameters $(\tau, \gamma)$ for the specific class of functions under consideration. Results from algebraic geometry (e.g., [NZ03]) suggest that the explicit computation of these parameters might be possible for large families of functions, such as the family of piecewise linear functions. This latter family of functions is particularly of interest, as deep nets with rectifier nonlinearities are piecewise affine. Identifying an upper bound on the adversarial robustness of deep nets in terms of the *depth* of the network would be a great step towards having a better understanding of such systems. In particular, this would allow us (just like in the linear case) to determine how far is the robustness of current deep nets from the maximum *achievable* robustness. Establishing sufficiently small upper bounds on the robustness of such families of classifiers would show that the current instability to adversarial perturbations of deep nets is more related to the *architecture*, rather than to the *learning algorithm* used for the training. In other words, the design of robust classifiers would then necessarily come from a "complexification" of the current deep architectures, just like we

were able to obtain more robust classifiers by increasing the degree of polynomial classifiers.

In general, one of the fundamental goals of this thesis is to provide an analysis of the robustness properties of classifiers, in the perspective of improving the classifiers' robustness to various forms of perturbations that might appear in practical systems. While many recent works have proposed *learning* methods for improving the robustness (e.g., [GR14; LHL15; Hua+15]), it would be interesting to look for different *architectures* that improve the robustness. In the future, we would like to use the insights derived in this thesis in order to propose new robust classification architectures. For example, this can come by imposing geometric constraints on the decision boundaries of the classifier, as we have shown in Chapter 5 that classifiers with sufficiently small curvature have a small robustness to semi-random noise. Hence, by constraining the decision boundaries of classifiers to have a large curvature, we might be able to achieve higher robustness to perturbations. Note however that enforcing geometric constraints on the decision boundary is not an easy task, as the decision boundary cannot be expressed in closed-form for most interesting classifiers, and one would therefore need to find surrogates to the curvature of the decision boundary.

Another line of research is to develop methods to visualize the decision boundary of high-dimensional classifiers. While we provided in Chapter 5 a simple way to visualize the decision boundary through the projection of the boundary on a two-dimensional plane, it is important to derive more systematic and complete methods that take into account the complexity of the classifier's decision boundary. Visualizing the effect of the classifier in the input space is certainly an important problem that would help us have a better understanding of the classification models that are often treated as black boxes. Of particular interest is how the classifier partitions the input space into different *regions*, where a region is a subset of the input space where the estimated label is constant. It should however be noted that the high dimensionality of most interesting classification problems (e.g., ImageNet) is a major challenge to developing such insightful visualization tools. In particular, the preservation of the geometry of the input space (to some extent) is an important requirement that makes the development of such tools challenging.

While we assumed in this thesis to have full knowledge of the classifier when computing adversarial examples, an interesting direction of research is to assess the robustness of classifiers to perturbations, when the adversary has only *limited* knowledge of the classification model. Our study of the robustness of classifiers to *semi*-random noise (Chapter 5) can be seen as a starting point for this analysis. In fact, we have shown that state-of-the-art classifiers are not robust to such noise that is only partially adversarial, and therefore does *not* require the full knowledge of the gradient of the classification function. The projection of the classifier gradients onto low-dimensional random subspaces is indeed sufficient to compute such perturbations. Building on this work, we believe it is possible to design attacks that use minimal information about the classifier $f$ (e.g., only few evaluations of $f$, but not the gradient). This can have many applications in the field of computer and mobile security as recently outlined in [Car+16].

Finally, while this thesis is mainly concerned with the problem of classification, we believe that the framework proposed in this thesis can be extended to various forms of machine learning tasks (e.g., detection, image segmentation). We believe this is a subject that is definitely worth investigation.

# A Appendix for Chapter 4

## A.1 Proof of Lemma 1

We begin by proving the following inequality:

**Lemma 4.** *Let $z_1, \dots, z_n$ be non-negative real numbers, and let $0 \leq \gamma \leq 1$. Then,*

$$\sum_{i=1}^{n} z_i^{\gamma} \leq n^{1-\gamma} \left( \sum_{i=1}^{n} z_i \right)^{\gamma}.$$

*Proof.* We prove that the quantity

$$\frac{\sum_{i=1}^{n} z_i^{\gamma}}{(\sum_{i=1}^{n} z_i)^{\gamma}} = \sum_{i=1}^{n} \left( \frac{z_i}{\sum_{i=1}^{n} z_i} \right)^{\gamma}$$

is bounded from above by $n^{1-\gamma}$. To do so, let $u_i = \frac{z_i}{\sum_{i=1}^{n} z_i}$, and let us determine the maximum of the concave function $g(u_1, \dots, u_{n-1}) = u_1^{\gamma} + \cdots + (1 - u_1 - \cdots - u_{n-1})^{\gamma}$. Setting the derivative of $g$ with respect to $u_i$ to zero, we get

$$u_i^{\gamma-1} - (1 - u_1 - \cdots - u_{n-1})^{\gamma-1} = 0,$$

hence $u_i = 1 - u_1 - \cdots - u_{n-1}$. We therefore get $u_1 = \cdots = u_{n-1}$, and conclude that the maximum of $\sum_{i=1}^{n} \left( \frac{z_i}{\sum_{i=1}^{n} z_i} \right)^{\gamma}$ is reached when $z_1 = \cdots = z_n$ and the value of the maximum is $n^{1-\gamma}$. □

We now prove Lemma 1, that we recall as follows:

**Lemma 1.** *Let $f$ be an arbitrary classifier that satisfies (A) with parameters $(\tau, \gamma)$. Then,*

$$\rho_{adv}(f) \leq 4^{1-\gamma} \tau \left( p_1 \mathbb{E}_{\mu_1}(f(\boldsymbol{x})) - p_{-1} \mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})) + 2\|f\|_{\infty} R(f) \right)^{\gamma}.$$

*Proof.* The goal is to find an upper bound on $\rho_{\text{adv}}(f) = \mathbb{E}_{\mu}(\Delta_{\text{adv}}(\boldsymbol{x}; f))$.

$$\begin{aligned}\rho_{\text{adv}}(f) &= p_1\mathbb{E}_{\mu_1}(\Delta_{\text{adv}}(\boldsymbol{x})) + p_{-1}\mathbb{E}_{\mu_{-1}}(\Delta_{\text{adv}}(\boldsymbol{x})) \\ &= p_1\Big(\mathbb{E}_{\mu_1}(\Delta_{\text{adv}}(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0)\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) \geq 0) + \mathbb{E}_{\mu_1}(\Delta_{\text{adv}}(\boldsymbol{x})|f(\boldsymbol{x}) < 0)\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) < 0)\Big) \\ &\quad + p_{-1}\Big(\mathbb{E}_{\mu_{-1}}(\Delta_{\text{adv}}(\boldsymbol{x})|f(\boldsymbol{x}) < 0)\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) < 0) + \mathbb{E}_{\mu_{-1}}(\Delta_{\text{adv}}(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0)\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) \geq 0)\Big).\end{aligned}$$

Using assumption (A), the following upper bounds hold:

$$\begin{aligned}&\mathbb{E}_{\mu_{\pm 1}}(\Delta_{\text{adv}}(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0) \leq \tau\mathbb{E}_{\mu_{\pm 1}}(f(\boldsymbol{x})^{\gamma}|f(\boldsymbol{x}) \geq 0) \\ &\mathbb{E}_{\mu_{\pm 1}}(\Delta_{\text{adv}}(\boldsymbol{x})|f(\boldsymbol{x}) < 0) \leq \tau\mathbb{E}_{\mu_{\pm 1}}((-f(\boldsymbol{x}))^{\gamma}|f(\boldsymbol{x}) < 0)\end{aligned}$$

Hence, we obtain the following inequality on $\rho_{\text{adv}}(f)$:

$$\begin{aligned}\rho_{\text{adv}}(f) &\leq \tau p_1\Big(\mathbb{E}_{\mu_1}(f(\boldsymbol{x})^{\gamma}|f(\boldsymbol{x}) \geq 0)\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) \geq 0) + \mathbb{E}_{\mu_1}((-f(\boldsymbol{x}))^{\gamma}|f(\boldsymbol{x}) < 0)\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) < 0)\Big) \\ &\quad + \tau p_{-1}\Big(\mathbb{E}_{\mu_{-1}}((-f(\boldsymbol{x}))^{\gamma}|f(\boldsymbol{x}) < 0)\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) < 0) + \mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})^{\gamma}|f(\boldsymbol{x}) \geq 0)\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) \geq 0)\Big).\end{aligned}$$

Using Jensen's inequality, we have $\mathbb{E}(X^{\gamma}) \leq \mathbb{E}(X)^{\gamma}$, for any random variable $X$, and $\gamma \leq 1$.
Using this inequality together with $\mathbb{P}(A) \leq \mathbb{P}(A)^{\gamma}$, we obtain

$$\begin{aligned}\rho_{\text{adv}}(f) \leq \tau\Big(&(p_1\mathbb{E}_{\mu_1}(f(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0)\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) \geq 0))^{\gamma} + (p_1\mathbb{E}_{\mu_1}(-f(\boldsymbol{x})|f(\boldsymbol{x}) < 0)\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) < 0))^{\gamma} \\ &+ \big(p_{-1}\mathbb{E}_{\mu_{-1}}(-f(\boldsymbol{x})|f(\boldsymbol{x}) < 0)\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) < 0)\big)^{\gamma} + \big(p_{-1}\mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0)\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) \geq 0)\big)^{\gamma}\Big).\end{aligned}$$

We use the result in Lemma 4 with $n = 4$, and obtain

$$\begin{aligned}\rho_{\text{adv}}(f) \leq \tau 4^{1-\gamma}\Big(&p_1\mathbb{E}_{\mu_1}(f(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0)\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) \geq 0) + p_1\mathbb{E}_{\mu_1}(-f(\boldsymbol{x})|f(\boldsymbol{x}) < 0)\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) < 0) \\ &+ p_{-1}\mathbb{E}_{\mu_{-1}}(-f(\boldsymbol{x})|f(\boldsymbol{x}) < 0)\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) < 0) + p_{-1}\mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0)\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) \geq 0)\Big)^{\gamma}.\end{aligned}$$

Note moreover that the following equality holds

$$\begin{aligned}&- p_1\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) < 0)\mathbb{E}_{\mu_1}(f(\boldsymbol{x})|f(\boldsymbol{x}) < 0) \\ &= 2p_1\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) < 0)|\mathbb{E}_{\mu_1}(f(\boldsymbol{x})|f(\boldsymbol{x}) < 0)| + p_1\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) < 0)\mathbb{E}_{\mu_1}(f(\boldsymbol{x})|f(\boldsymbol{x}) < 0),\end{aligned}$$

Using the above equality along with a similar one for $p_{-1}\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) \geq 0)\mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0)$, the following upper bound is obtained

$$\begin{aligned}\rho_{\text{adv}}(f) \leq \tau 4^{1-\gamma}\Big(&p_1\mathbb{E}_{\mu_1}(f(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0)\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) \geq 0) + p_1\mathbb{E}_{\mu_1}(f(\boldsymbol{x})|f(\boldsymbol{x}) < 0)\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) < 0) \\ &- p_{-1}\mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})|f(\boldsymbol{x}) < 0)\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) < 0) - p_{-1}\mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0)\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) \geq 0) \\ &+ 2p_1\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) < 0)|\mathbb{E}_{\mu_1}(f(\boldsymbol{x})|f(\boldsymbol{x}) < 0| + 2p_{-1}\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) \geq 0)|\mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0|)\Big)^{\gamma},\end{aligned}$$

which simplifies to

$$\rho_{\text{adv}}(f) \leq \tau 4^{1-\gamma}\Big(p_1\mathbb{E}_{\mu_1}(f(\boldsymbol{x})) - p_{-1}\mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})) + 2p_1\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) < 0)|\mathbb{E}_{\mu_1}(f(\boldsymbol{x})|f(\boldsymbol{x}) < 0)| + 2p_{-1}\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) \geq 0)|\mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0)|\Big)^{\gamma},$$

Observe moreover that $R(f) = p_1\mathbb{P}_{\mu_1}(f(\boldsymbol{x}) < 0) + p_{-1}\mathbb{P}_{\mu_{-1}}(f(\boldsymbol{x}) \geq 0)$, and that $|\mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})|f(\boldsymbol{x}) \geq 0)|$ is bounded from above by $\|f\|_\infty$. We therefore conclude that

$$\rho_{\text{adv}}(f) \leq \tau 4^{1-\gamma}\Big(p_1\mathbb{E}_{\mu_1}(f(\boldsymbol{x})) - p_{-1}\mathbb{E}_{\mu_{-1}}(f(\boldsymbol{x})) + 2R(f)\|f\|_\infty\Big)^{\gamma}.$$

□

## A.2 Discussion on the norms used to measure the magnitude of adversarial perturbations

The goal of this section is to discuss different ways of measuring the robustness to adversarial perturbations.

Given a datapoint $\boldsymbol{x}$, let $\eta > 0$ be such that we know a priori that all points in the region

$$\mathcal{R}(\boldsymbol{x}) = \{\boldsymbol{z} : N(\boldsymbol{z} - \boldsymbol{x}) \leq \eta\},$$

have the same true class as $\boldsymbol{x}$ (i.e., a human observer would classify all images in this region similarly). Here $N : \mathbb{R}^d \to \mathbb{R}^+$ defines a norm in the image space. Note that $\mathcal{R}(\boldsymbol{x})$ only depends on the dataset, but does not depend on any classifier $f$. We defined the robustness of $f$ to adversarial perturbations, at $\boldsymbol{x}$, to be

$$\Delta_{\text{adv}}(\boldsymbol{x}) = \min_{\boldsymbol{r}} N(\boldsymbol{r}) \text{ subject to } f(\boldsymbol{x}+\boldsymbol{r})f(\boldsymbol{x}) \leq 0.$$

The classifier $f$ is said to be *not robust* at $\boldsymbol{x}$ if

$$\Delta_{\text{adv}}(\boldsymbol{x}) \leq \eta. \tag{A.1}$$

In words, this means that there exists a point $\boldsymbol{z}$ in the region $\mathcal{R}(\boldsymbol{x})$ (i.e., $\boldsymbol{z}$ and $\boldsymbol{x}$ are classified in the same way by a human observer), but $\boldsymbol{z}$ is classified differently than $\boldsymbol{x}$ by $f$. Our main theoretical result provides upper bounds to $\rho_{\text{adv}}(f)$ (the expectation of $\Delta_{\text{adv}}(\boldsymbol{x})$) in terms of interpretable quantities (i.e., distinguishability and risk): $\rho_{\text{adv}}(f) \leq U(\mu, R(f))$. Using this upper bound and Eq. (A.1), we certify that $f$ is *not* robust to adversarial perturbations when the following sufficient condition holds:

$$U(\mu, R(f)) \leq \eta. \tag{A.2}$$

The main difficulty in the above definitions lies in the choice of $N$ and $\eta$: how can $(N, \eta)$ be chosen to guarantee that $\mathcal{R}(\boldsymbol{x})$ contains all images of the same underlying class as $\boldsymbol{x}$? In the original paper [Sze+14], $N$ is set to be the $\ell_2$ norm, but no $\eta$ is formally

derived; classifiers are said to be not robust to adversarial perturbations when $\rho_{\text{adv}}(f)/\sqrt{d}$ is judged to be "sufficiently small". For example, it appears from Table 1 in [Sze+14] that if $\rho_{\text{adv}}(f)/\sqrt{d} \lesssim 0.1$, the minimum required perturbation is thought to be small enough to guarantee that the images do not change their true underlying label. Motivated by the fact that pixels (or features) have limited precision, [GSS15] consider instead the $\ell_\infty$ norm, and ideally assume that a perturbation that have $\ell_\infty$ norm smaller than the precision of the pixels (e.g., 1/255 of the dynamic range for 8-bit images) is guaranteed to conserve the true underlying class. While this corresponds to setting $\eta$ to be the precision of the pixels, in practice it is set to be much larger for the MNIST case, as the images are essentially binary. In our case, the $\ell_2$ norm is considered, and we define the quantity $\rho_d$ to be the average norm of the minimal perturbation required to transform a training point to a training point of the opposite class[1]:

$$\rho_d = \frac{1}{m}\sum_{i=1}^{m} \min_{j:y(\boldsymbol{x}_j)\neq y(\boldsymbol{x}_i)} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2.$$

We assume that the image $\boldsymbol{x}+\boldsymbol{r}$ is of the same underlying label as $\boldsymbol{x}$ if $\|\boldsymbol{r}\|_2$ is one order of magnitude smaller than $\rho_d$. This corresponds to setting $\eta = \rho_d/10$. A summary of the different choices is shown in Table A.1.

| | $N$ | $\eta$ |
|---|---|---|
| [Sze+14] | $\|\cdot\|_2$ | - |
| [GSS15] | $\|\cdot\|_\infty$ | Determined by the image precision (in theory). Larger in practice. |
| Ours | $\|\cdot\|_2$ | $\rho_d/10$ |

Table A.1: Different choices of $N$ and $\eta$ in different papers.

All the above choices represent proxies of what we really would like to capture (i.e., the notion of perceptibility and class change). They all have some benefits and drawbacks, which we mention briefly now. We first acknowledge that the $\ell_\infty$ norm with $\eta \approx 0.1$ blocks class changes (and therefore provides a sufficient condition for certifying the non-robustness of classifiers) for images that are *essentially binary* (e.g, MNIST digit images). In those cases, the $\ell_\infty$ norm seems more appropriate to use than the $\ell_2$ norm. However, in order to compare both norms, we need to carefully (and fairly) choose the $\eta$ parameter for both norms. In fact, if it is acknowledged that $N = \|\cdot\|_\infty$ and $\eta = 0.1$ provides a valid region $\mathcal{R}$ where underlying image classes do not change, then $N = \|\cdot\|_2$ and $\eta = 0.1$ *also provides a valid region*, as $\|\boldsymbol{r}\|_\infty \leq \|\boldsymbol{r}\|_2$ for any vector $\boldsymbol{r}$. It is therefore all a matter of choosing a right threshold $\eta$ that is fair for all norms, if we wish to compare the norms for the task that we have at hand. A comparison between the $\ell_\infty$ and $\ell_2$ norm is provided in [Goo15], and it is concluded that, while the $\ell_2$ norm allows for class changes within its region, the $\ell_\infty$ essentially blocks the class changes and therefore constitutes a better choice. In more details, the comparison goes as follows: it is first argued that by choosing $N = \|\cdot\|_2$ and $\eta = 3.96$, the region $\mathcal{R}$ contains both a "natural" 3 and 7, and therefore does not provide a valid region. To show the benefits of the $\ell_\infty$ norm, the author proceeds by considering $N = \|\cdot\|_\infty$ and $\eta = 3.96/\sqrt{d} \approx 0.1414$. It is then argued that this region blocks previous attempts for class changes, and therefore the $\ell_\infty$ norm provides a better choice for the task at hand. While this type of comparison is important in order to reach a better understanding of the

[1] We consider here a non-normalized version of the data robustness $\rho_d$ for simplicity of the exposition.

norms used to measure the adversarial examples, it is not conclusive as it is unfair to the $\ell_2$ norm. Let us recall the following inequalities

$$\forall \boldsymbol{r} \in \mathbb{R}^d, \|\boldsymbol{r}\|_\infty \leq \|\boldsymbol{r}\|_2 \leq \sqrt{d}\|\boldsymbol{r}\|_\infty. \tag{A.3}$$

For a fixed $\eta_0 > 0$, define the regions:

$$\mathcal{R}_\infty = \{\boldsymbol{z} : \|\boldsymbol{z} - \boldsymbol{x}\|_\infty \leq \eta_0\},$$
$$\mathcal{R}_2 = \{\boldsymbol{z} : \|\boldsymbol{z} - \boldsymbol{x}\|_2 \leq \eta_0\sqrt{d}\}.$$

It should be noted that for any $\eta_0$, we have $\mathcal{R}_\infty \subset \mathcal{R}_2$ using Eq. (A.3). Not only that, but $\mathcal{R}_\infty$ constitutes *a tiny portion* of $\mathcal{R}_2$ in high dimensional spaces (i.e., the volume of $\mathcal{R}_\infty$ over that of $\mathcal{R}_2$ decays exponentially with the dimension). Therefore, a comparison of $\mathcal{R}_2$ to $\mathcal{R}_\infty$ will typically lead to problematic images in $\mathcal{R}_2$ but not in $\mathcal{R}_\infty$, as $\mathcal{R}_2$ is much bigger than $\mathcal{R}_\infty$. Therefore, the fact that $\mathcal{R}_\infty$ is a much smaller set than $\mathcal{R}_2$ (i.e., it contains much less images) is already known from Eq. (A.3) and is not conclusive in terms of the comparison of the two norms for measuring the robustness to adversarial perturbations. Just like the comparison of $\mathcal{R}_2$ to $\mathcal{R}_\infty$ is unfair to the $\ell_2$ norm, saying that the $\ell_2$ norm is better than the $\ell_\infty$ norm because $\mathcal{R}_\infty$ contains much more images (potentially problematic ones with class changes, for sufficiently large $\eta_0$) that are not in $\mathcal{R}'_2 = \{\boldsymbol{z} : \|\boldsymbol{z} - \boldsymbol{x}\|_2 \leq \eta_0\}$ is unfair to the $\ell_\infty$ norm.

One possible way for providing a fair comparison between both norms is to find the coefficient $c$ such that $\mathcal{R}_\infty$ has the *same volume* as the following $\ell_2$ ball

$$\mathcal{R}''_2 = \{\boldsymbol{z} : \|\boldsymbol{z} - \boldsymbol{x}\|_2 \leq \eta''_0\sqrt{d}\}, \text{ with } \eta''_0 = \eta_0 c.$$

Using mathematical derivations that we omit for the flow of this short discussion, we obtain $c = \sqrt{\frac{2}{e\pi}} \approx 0.48$ asymptotically as $d \to \infty$. We argue that the comparison of $\mathcal{R}_\infty$ to $\mathcal{R}''_2$ provides a more conclusive experiment than comparing $\mathcal{R}_\infty$ to $\mathcal{R}_2$, as it highlights the advantage of one norm with respect to the other without biases on the volume of the region. In practice, this new comparison implies the following change for the "3" vs. "7" example in [Goo15]: instead of allowing perturbations of max-norm 0.1414, perturbations with $\ell_\infty$ norm up to $\approx 0.3$ are allowed. This will result in images that are roughly twice as much perturbed, for the $\ell_\infty$ case. Even with this comparison, it is possible that the max-norm in this case will also block attempts to change the class, as the images are essentially binary. We believe that the $\ell_\infty$ is probably a better choice in this case.

However, this is *not* a general statement, as in some cases of non-binary images, the $\ell_2$ norm might be a better choice. We illustrate the above statement on a toy example where the goal is to classify sport balls. Some example images are shown in Fig. A.1. In this example, the $\ell_\infty$ norm between any two images is less than 0.11. Setting $\eta_0 = 0.11$ with $N = \|\cdot\|_\infty$ does not define a valid region (i.e., it does *not* guarantee that no class changes will occur within the region). On the other hand, the region $\mathcal{R}''_2$ computed with $\eta_0 = 0.11$ (i.e., $\eta''_0 = 0.0532$) rightfully excludes the images b) and c) from the space of valid perturbations of a). This toy example provides a proof of concept that shows that, in some cases, the $\ell_2$ norm might actually be a better choice than the $\ell_\infty$ norm.

(a) (b) (c)

Figure A.1: Example images in a toy classification problem where the goal is to distinguish the different balls (a: basketball, b: soccer). (c) represents an umbrella that does not belong to any class. Black pixels are equal to 0, white pixels are equal to 1, grey pixels are set around 0.9.

In conclusion, we stress that this example has no intention of proving that the $\ell_2$ norm is universally better than the $\ell_\infty$ norm to measure the norm of adversarial perturbations. Through this discussion and example, we show that there is no universal answer to which norm one has to use to measure the robustness to adversarial perturbations, as it is strongly application-dependent. We believe more theoretical research in that area is needed in order to fully grasp the advantage of each norm, and probably design new norms that are suitable for measuring adversarial perturbations.

# B Appendix for Chapter 5

## B.1 Proof of Theorem 3 (affine classifiers)

**Lemma 5** ([DG03]). *Let $Y$ be a point chosen uniformly at random from the surface of the $d$-dimensional sphere $\mathbb{S}^{d-1}$. Let the vector $Z$ be the projection of $Y$ onto its first $m$ coordinates, with $m < d$. Then,*

1. *If $\beta < 1$, then*

$$\mathbb{P}\left(\|Z\|_2^2 \leq \frac{\beta m}{d}\right) \leq \beta^{m/2}\left(1 + \frac{(1-\beta)m}{(d-m)}\right)^{(d-m)/2} \leq \exp\left(\frac{m}{2}(1-\beta+\ln\beta)\right). \tag{B.1}$$

2. *If $\beta > 1$, then*

$$\mathbb{P}\left(\|Z\|_2^2 \geq \frac{\beta m}{d}\right) \leq \beta^{m/2}\left(1 + \frac{(1-\beta)m}{(d-m)}\right)^{(d-m)/2} \leq \exp\left(\frac{m}{2}\left(1-\beta+\ln\beta\right)\right). \tag{B.2}$$

**Lemma 6.** *Let $\boldsymbol{v}$ be a random vector uniformly drawn from the unit sphere $\mathbb{S}^{d-1}$, and $\mathbf{P}_m$ be the projection matrix onto the first $m$ coordinates. Then,*

$$\mathbb{P}\left(\beta_1(\delta,m)\frac{m}{d} \leq \|\mathbf{P}_m\boldsymbol{v}\|_2^2 \leq \beta_2(\delta,m)\frac{m}{d}\right) \geq 1-2\delta, \tag{B.3}$$

*with $\beta_1(\delta,m) = \max((1/e)\delta^{2/m}, 1-\sqrt{2(1-\delta^{2/m})}$, and $\beta_2(\delta,m) = 1 + 2\sqrt{\frac{\ln(1/\delta)}{m}} + \frac{2\ln(1/\delta)}{m}$.*

*Proof.* Note first that the upper bound of Lemma 5 can be bounded as follows:

$$\beta^{m/2}\left(1 + \frac{(1-\beta)m}{d-m}\right)^{(d-m)/2} \leq \beta^{m/2}\exp\left(\frac{(1-\beta)m}{2}\right), \tag{B.4}$$

using $1 + x \leq \exp(x)$. We find $\beta$ such that $\beta^{m/2}\exp\left(\frac{(1-\beta)m}{2}\right) \leq \delta$, or equivalently, $\beta\exp(1-\beta) \leq \delta^{2/m}$. It is easy to see that when $\beta = \frac{1}{e}\delta^{2/m}$, the inequality holds. Note however that $\frac{1}{e}\delta^{2/m}$ does not converge to 1 as $m \to \infty$. We therefore need to derive a tighter bound for this regime. Using the inequality $\beta\exp(1-\beta) \leq 1 - \frac{1}{2}(1-\beta)^2$ for $0 \leq \beta \leq 1$, it

follows that the inequality $\beta \exp(1-\beta) \leq \delta^{2/m}$ holds for $\beta = 1 - \sqrt{2(1-\delta^{2/m})}$. In this case, we have $1 - \sqrt{2(1-\delta^{2/m})} \to 1$, as $m \to \infty$. We take our lower bound to be the max of both derived bounds (the latter is more appropriate for large $m$, whereas the former is tighter for small $m$).

For $\beta_2$, note that the requirement $\beta \exp(1-\beta) \leq \delta^{2/m}$ is equivalent to $-\ln(\beta) + (\beta - 1) \geq \frac{2}{m}\ln(1/\delta)$. By setting $\beta = \beta_2(\delta, m)$, this condition is equivalent to $2\sqrt{\frac{\ln(1/\delta)}{m}} - \ln(\beta_2(\delta, m)) \geq 0$, or equivalently, $2z - \ln(1 + 2z + 2z^2) \geq 0$, with $z = \sqrt{\frac{\ln(1/\delta)}{m}}$. The function $z \mapsto 2z - \ln(1 + 2z + 2z^2) \geq 0$ is positive on $\mathbb{R}^+$. Hence, $\beta_2(\delta, m)$ satisfies $\beta \exp(1-\beta) \leq \delta^{2/m}$, which concludes the proof. □

We now prove our main theorem for linear classifiers that we recall as follows:

**Theorem 3.** *Let $\mathcal{S}$ be a random $m$-dimensional subspace of $\mathbb{R}^d$. The following inequalities hold between the norms of semi-random perturbation $\boldsymbol{r}^*_{\mathcal{S}}$ and the worst-case perturbation $\boldsymbol{r}^*$. Let $\zeta_1(m,\delta) = \frac{1}{\beta_2(m,\delta)}$, and $\zeta_2(m,\delta) = \frac{1}{\beta_1(m,\delta)}$.*

$$\zeta_1(m,\delta)\frac{d}{m}\|\boldsymbol{r}^*\|_2^2 \leq \|\boldsymbol{r}^*_{\mathcal{S}}\|_2^2 \leq \zeta_2(m,\delta)\frac{d}{m}\|\boldsymbol{r}^*\|_2^2, \tag{B.5}$$

*with probability exceeding $1 - 2(L+1)\delta$.*

*Proof.* For the linear case, $\boldsymbol{r}^*$ and $\boldsymbol{r}^*_{\mathcal{S}}$ can be computed in closed form. We recall that (Fact 5), for any subspace $\mathcal{S}$, we have

$$\boldsymbol{r}^k_{\mathcal{S}} = \frac{\left|f_k(\boldsymbol{x}_0) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0)\right|}{\|\mathbf{P}_{\mathcal{S}}\boldsymbol{w}_k - \mathbf{P}_{\mathcal{S}}\boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}\|_2^2}(\mathbf{P}_{\mathcal{S}}\boldsymbol{w}_k - \mathbf{P}_{\mathcal{S}}\boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}), \tag{B.6}$$

where $\boldsymbol{r}^k_{\mathcal{S}}$ was defined in Eq. (5.9). In particular, when $\mathcal{S} = \mathbb{R}^d$, we have

$$\boldsymbol{r}^k = \frac{\left|f_k(\boldsymbol{x}_0) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0)\right|}{\|\boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}\|_2^2}(\boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}). \tag{B.7}$$

Let $k \neq \hat{k}(\boldsymbol{x}_0)$. Define, for the sake of readability

$$f^k = \left|f_k(\boldsymbol{x}_0) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0)\right|,$$
$$\boldsymbol{z}^k = \boldsymbol{w}_k - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}.$$

Note that

$$\frac{\|\boldsymbol{r}^k\|_2^2}{\|\boldsymbol{r}^k_{\mathcal{S}}\|_2^2} = \frac{\|\mathbf{P}_{\mathcal{S}}\boldsymbol{z}^k\|_2^2}{\|\boldsymbol{z}^k\|_2^2}. \tag{B.8}$$

The projection of a fixed vector in $\mathbb{S}^{d-1}$ onto a random $m$ dimensional subspace is equivalent (up to a unitary transformation $\mathbf{U}$) to the projection of a random vector uniformly sampled from $\mathbb{S}^{d-1}$ into a fixed subspace. Let $\mathbf{P}_m$ be the projection onto the first $m$ coordinates.

Observe that

$$\|\mathbf{P}_{\mathcal{S}}\boldsymbol{z}^k\|_2^2 = \|\mathbf{U}^T\mathbf{P}_m\mathbf{U}\boldsymbol{z}^k\|_2^2 = \|\mathbf{P}_m\mathbf{U}\boldsymbol{z}^k\|_2, \tag{B.9}$$

Hence, we have

$$\frac{\|\mathbf{P}_{\mathcal{S}}\boldsymbol{z}^k\|_2^2}{\|\boldsymbol{z}^k\|_2^2} = \|\mathbf{P}_m\boldsymbol{y}\|_2^2, \tag{B.10}$$

where $\boldsymbol{y}$ is a random vector distributed uniformly in the unit sphere $\mathbb{S}^{d-1}$. We apply Lemma 6, and obtain

$$\mathbb{P}\left(\beta_1(m,\delta)\frac{m}{d} \leq \|\mathbf{P}_m\boldsymbol{y}\|_2^2 \leq \beta_2(m,\delta)\frac{m}{d}\right) \geq 1-2\delta. \tag{B.11}$$

Hence,

$$\mathbb{P}\left\{\frac{1}{\beta_2(m,\delta)}\frac{d}{m} \leq \frac{\|\boldsymbol{r}_{\mathcal{S}}^k\|_2^2}{\|\boldsymbol{r}^k\|_2^2} \leq \frac{1}{\beta_1(m,\delta)}\frac{d}{m}\right\} \geq 1-2\delta. \tag{B.12}$$

Using the multi-class extension in Lemma 7, we conclude that

$$\mathbb{P}\left\{\zeta_1(m,\delta)\frac{d}{m} \leq \frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2^2}{\|\boldsymbol{r}^*\|_2^2} \leq \zeta_2(m,\delta)\frac{d}{m}\right\} \geq 1-2(L+1)\delta. \tag{B.13}$$

□

**Lemma 7** (Binary case to multiclass)**.** *Assume that, for all* $k \in \{1,\dots,L]\}\backslash\{\hat{k}(\boldsymbol{x}_0)\}$

$$\mathbb{P}\left(l \leq \frac{\|\boldsymbol{r}_{\mathcal{S}}^k\|_2}{\|\boldsymbol{r}^k\|_2} \leq u\right) \geq 1-\delta. \tag{B.14}$$

*Then, we have*

$$\mathbb{P}\left(l \leq \frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq u\right) \geq 1-(L+1)\delta. \tag{B.15}$$

*Proof.* Let $p := \arg\min_i \|\boldsymbol{r}^i\|_2$. Note that we have $\mathbb{P}\left(\frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \geq u\right) \leq \mathbb{P}\left(\frac{\|\boldsymbol{r}_{\mathcal{S}}^p\|_2}{\|\boldsymbol{r}^p\|_2} \geq u\right) \leq \delta$. Moreover, we use a union bound to bound the the other bad event probability:

$$\mathbb{P}\left(\frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq l\right) \leq \mathbb{P}\left(\bigcup_k \left\{\frac{\|\boldsymbol{r}_{\mathcal{S}}^k\|_2}{\|\boldsymbol{r}^k\|_2} \leq l\right\}\right) \leq L\delta, \tag{B.16}$$

$$\tag{B.17}$$

We conclude by using the fact that

$$\mathbb{P}\left(l \leq \frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq u\right) = 1 - \mathbb{P}\left(\frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq l\right) - \mathbb{P}\left(\frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \geq u\right). \tag{B.18}$$

□

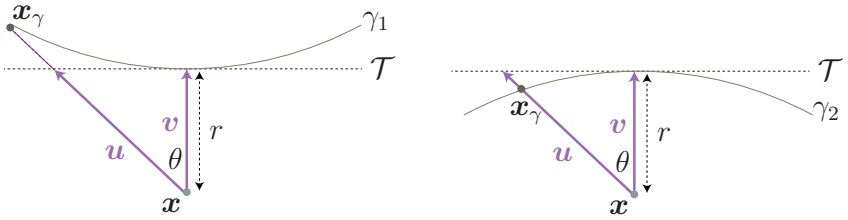



Figure B.1: Bounding $\|\boldsymbol{x}_\gamma - \boldsymbol{x}\|_2$ in terms of $\kappa$.

## B.2 Proof of Theorem 4 and Corollary 1 (nonlinear classifiers)

First, we present an important geometric lemma and then use it to bound $\|\boldsymbol{r}^*_{\mathcal{S}}\|_2$. For the sake of the general readability of the section, some auxiliary results are given in Section B.2.1.

In the following result, we show that, when the curvature of a planar curve is constant and sufficiently small, the distance between a point $\boldsymbol{x}$ and the curve at a specific direction $\theta$ is well approximated by the distance between $\boldsymbol{x}$ and a straight line (see Figure B.1 for an illustration).

**Lemma 8.** *Let $\gamma$ be a planar curve of constant curvature $\kappa$. We denote by $r$ the distance between a point $\boldsymbol{x}$ and the curve $\gamma$. Denote moreover by $\mathcal{T}$ the tangent to $\gamma$ at the closest point to $\boldsymbol{x}$ (see Figure B.1). Let $\theta$ be the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$ as depicted in Figure B.1. We assume that $r\kappa < 1$. We have*

$$-C_1 r\kappa \tan^2(\theta) \leq \frac{\|\boldsymbol{x}_\gamma - \boldsymbol{x}\|_2}{\|\boldsymbol{u}\|_2} - 1 \tag{B.19}$$

*Moreover, if*

$$\tan^2(\theta) \leq \frac{0.2}{r\kappa},$$

*then, the following upper bound holds*

$$\frac{\|\boldsymbol{x}_\gamma - \boldsymbol{x}\|_2}{\|\boldsymbol{u}\|_2} - 1 \leq C_2 r\kappa \tan^2(\theta). \tag{B.20}$$

*We can set $C_1 = 0.625$ and $C_2 = 2.25$.*

*Proof of upper bound.* We consider two distinct cases for the curve $\gamma$. In the case where $\gamma$ is concave-shaped (Fig. B.1, right figure), we have

$$\frac{\|\boldsymbol{x}_\gamma - \boldsymbol{x}\|_2}{\|\boldsymbol{u}\|_2} \leq 1,$$

and the upper bound in Eq. (B.20) directly holds. We therefore focus on the case where $\gamma$ is convex-shaped as illustrated in the left figure of Fig. B.1. Define $R := 1/\kappa$, one can write using simple geometric inspection

$$R^2 = \sin(\theta) r'^2 + (R + r - r'\cos(\theta))^2, \tag{B.21}$$

where $r' = \|\boldsymbol{x}_\gamma - \boldsymbol{x}\|_2$. The discriminant of the second order equation (with variable $r'$) is equal to

$$\Delta = 4\left((R+r)^2\cos^2(\theta) - (2rR + r^2)\right).$$

We have $\Delta \geq 0$ as $\theta$ satisfies the two assumptions $\tan^2(\theta) \leq 0.2R/r$ and $r/R < 1$. The smallest solution of this second order equation is given as follows

$$r' = (R+r)\cos(\theta) - \sqrt{(R+r)^2\cos^2(\theta) - 2Rr - r^2}. \tag{B.22}$$

Using some simple algebraic manipulations, we obtain

$$r' = \frac{r}{\cos(\theta)}\left(\left(\frac{R}{r}+1\right)\cos^2(\theta) - \frac{R}{r}\cos^2(\theta)\sqrt{1 - \tan^2(\theta)\frac{2Rr+r^2}{R^2}}\right). \tag{B.23}$$

Using the inequality in Lemma 11 together with the two assumptions, we get

$$\begin{aligned} r' \leq \frac{r}{\cos(\theta)}\Bigg(\cos^2(\theta) + \frac{R}{r}\cos^2(\theta)\tan^2(\theta)\left(\frac{2Rr+r^2}{2R^2}\right) \\ + \frac{R}{r}\cos^2(\theta)\tan^4(\theta)\left(\frac{2Rr+r^2}{2R^2}\right)^2\Bigg). \end{aligned} \tag{B.24}$$

With simple trigonometric identities, the above expression can be simplified to

$$r' \leq \frac{r}{\cos(\theta)}\left(1 + \frac{r}{R}\left(\frac{\sin^2(\theta)}{2} + \frac{\sin^4(\theta)}{\cos^2(\theta)}\left(1+\frac{r}{2R}\right)^2\right)\right). \tag{B.25}$$

We expand this quantity, and obtain

$$r' \leq \frac{r}{\cos(\theta)}\left(1 + \left(\frac{\sin^2(\theta)}{2} + \frac{\sin^4(\theta)}{\cos^2(\theta)}\right)\frac{r}{R} + \frac{\sin^4(\theta)}{\cos^2(\theta)}\frac{r^2}{R^2} + \frac{\sin^4(\theta)}{4\cos^2(\theta)}\frac{r^3}{R^3}\right). \tag{B.26}$$

Since $\sin^2(\theta)\tan^2(\theta) = \tan^2(\theta) - \sin^2(\theta)$, we have

$$r' \leq \frac{r}{\cos(\theta)}\left(1 + \tan^2(\theta)\left(\frac{r}{R} + \frac{r^2}{R^2} + \frac{r^3}{4R^3}\right)\right). \tag{B.27}$$

According to the assumptions $r/R < 1$, therefore

$$r' \leq \frac{r}{\cos(\theta)}\left(1 + 2.25\tan^2(\theta)\frac{r}{R}\right). \tag{B.28}$$

Since $r/\cos(\theta) = \|\boldsymbol{u}\|_2$, one can finally conclude on the upper bound

$$\frac{\|\boldsymbol{x}_\gamma - \boldsymbol{x}\|_2}{\|\boldsymbol{u}\|_2} - 1 \leq 2.25r\kappa\tan^2(\theta). \tag{B.29}$$

□

*Proof of lower bound.* When the curve is convex shaped (Fig. B.1 left), we have $\|\boldsymbol{x}_\gamma - \boldsymbol{x}\|_2 \geq \|\boldsymbol{u}\|_2$, and the desired lower bound holds. We focus therefore on the case where $\gamma$ has a

concave shape, and coincides with with $\gamma_2$ (see Fig. B.1 right). The following equation holds using simple geometric arguments

$$R^2 = \sin(\theta)r'^2 + (R - r + r'\cos(\theta))^2. \tag{B.30}$$

where $r' = \|\boldsymbol{x}_\gamma - \boldsymbol{x}\|_2$. Solving this second order equation gives

$$r' = -(R-r)\cos(\theta) + \sqrt{(R-r)^2\cos^2(\theta) - r^2 + 2Rr}. \tag{B.31}$$

After some algebraic manipulations, we get

$$r' = \frac{r}{\cos(\theta)}\left(-\left(\frac{R}{r} - 1\right)\cos^2(\theta) + \frac{R}{r}\cos^2(\theta)\sqrt{1 + \tan^2(\theta)\frac{2Rr - r^2}{R^2}}\right). \tag{B.32}$$

Using the inequality in Lemma 12, together with the fact that $r\kappa < 1$, we obtain

$$\begin{aligned} r' \geq \frac{r}{\cos(\theta)}\Bigg(\cos^2(\theta) + \frac{R}{r}\cos^2(\theta)\tan^2(\theta)\left(\frac{2Rr - r^2}{2R^2}\right) \\ - \frac{R}{r}\frac{\cos^2(\theta)\tan^4(\theta)}{2}\left(\frac{2Rr - r^2}{2R^2}\right)^2\Bigg). \end{aligned} \tag{B.33}$$

Using simple trigonometric identities, the above expression is simplified to

$$r' \geq \frac{r}{\cos(\theta)}\left(1 + \frac{r}{R}\left(-\frac{\sin^2(\theta)}{2} - \frac{\sin^4(\theta)}{2\cos^2(\theta)}\left(1 - \frac{r}{2R}\right)^2\right)\right). \tag{B.34}$$

When expanding it, we obtain

$$r' \geq \frac{r}{\cos(\theta)}\left(1 - \left(\frac{\sin^2(\theta)}{2} + \frac{\sin^4(\theta)}{2\cos^2(\theta)}\right)\frac{r}{R} + \frac{\sin^4(\theta)}{2\cos^2(\theta)}\frac{r^2}{R^2} - \frac{\sin^4(\theta)}{8\cos^2(\theta)}\frac{r^3}{R^3}\right). \tag{B.35}$$

Since $\sin^2(\theta)\tan^2(\theta) = \tan^2(\theta) - \sin^2(\theta)$, we have

$$r' \geq \frac{r}{\cos(\theta)}\left(1 - \tan^2(\theta)\left(\frac{r}{2R} + \frac{r^3}{8R^3}\right)\right). \tag{B.36}$$

Using again the assumption $r/R < 1$, we obtain

$$r' \geq \frac{r}{\cos(\theta)}\left(1 - 0.625\tan^2(\theta)\frac{r}{R}\right). \tag{B.37}$$

Since $r/\cos(\theta) = \|\boldsymbol{u}\|_2$, one can rewrite it as

$$\frac{\|\boldsymbol{x}_\gamma - \boldsymbol{x}\|_2}{\|\boldsymbol{u}\|_2} - 1 \geq -0.625r\kappa\tan^2(\theta), \tag{B.38}$$

which completes the proof. □

We now use the previous lemma to bound the semi-random robustness of the classifier, i.e. $\|\boldsymbol{r}^k_{\mathcal{S}}\|_2$, to the worst-case robustness $\|\boldsymbol{r}^k\|_2$ in the case where the curvature is sufficiently small.

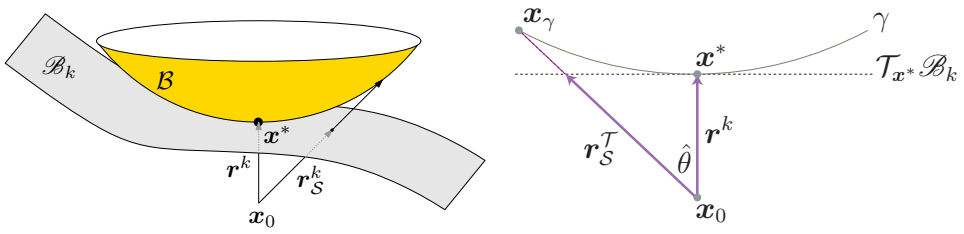



Figure B.2: Left: To prove the upper bound, we consider a ball $\mathcal{B}$ included in $\mathcal{R}_k$ that intersects with the boundary at $\boldsymbol{x}^*$. Upper bounds on $\|\boldsymbol{r}^k_{\mathcal{S}}\|_2$ derived when the boundary is $\partial\mathcal{B}$ are also valid upper bounds for the real boundary $\mathscr{B}_k$. Right: Normal section to the decision boundary $\mathscr{B}_k = \partial\mathcal{B}$ along the normal plane $\mathcal{U} = \text{span}\left(\boldsymbol{r}^{\mathcal{T}}_{\mathcal{S}}, \boldsymbol{r}^k\right)$. We denote by $\gamma$ the normal section of boundary $\mathscr{B}_k$, along the plane $\mathcal{U}$, and by $\mathcal{T}_{\boldsymbol{x}^*}\mathscr{B}_k$ the tangent space to the sphere $\partial\mathcal{B}$ at $\boldsymbol{x}^*$.

**Theorem 4.** *Let $\mathcal{S}$ be a random $m$-dimensional subspace of $\mathbb{R}^d$. Define $\alpha := \sqrt{m/d}$, and let $\kappa := \kappa(\mathscr{B}_k)$. Assuming that $\kappa \leq \frac{C\alpha^2}{\zeta_2(m,\delta)\|\boldsymbol{r}^k\|_2}$, the following inequalities hold between $\|\boldsymbol{r}^k_{\mathcal{S}}\|_2$ and the worst-case perturbation $\|\boldsymbol{r}^k\|_2$*

$$\frac{\zeta_1(m,\delta)}{\alpha^2}\left(1 - \frac{C_1\|\boldsymbol{r}^k\|_2\kappa\zeta_2(m,\delta)}{\alpha^2}\right)^2 \leq \frac{\|\boldsymbol{r}^k_{\mathcal{S}}\|_2^2}{\|\boldsymbol{r}^k\|_2^2} \leq \frac{\zeta_2(m,\delta)}{\alpha^2}\left(1 + \frac{C_2\|\boldsymbol{r}^k\|_2\kappa\zeta_2(m,\delta)}{\alpha^2}\right)^2 \tag{B.39}$$

*with probability larger than $1 - 4\delta$. The constants can be taken $C = 0.2, C_1 = 0.625, C_2 = 2.25$.*

*Proof of upper bound.* Denote by $\boldsymbol{x}^*$ the point belonging to the boundary $\mathscr{B}_k$ that is closest to the original data point $\boldsymbol{x}_0$. By definition of the curvature $\kappa$ (see Eq. 5.7), there exists a point $\boldsymbol{z}^*$ such that the ball $\mathcal{B}$ centered at $\boldsymbol{z}^*$ and of radius $1/\kappa = \|\boldsymbol{z}^* - \boldsymbol{x}^*\|_2$ is inscribed in the region $\mathcal{R}_k = \{x \in \mathbb{R}^d : f_k(\boldsymbol{x}) > f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x})\}$ (see Fig. B.2 (a)).[1]

Observe that the worst-case perturbation along any subspace $\mathcal{S}$ that reaches the ball $\mathcal{B}$ is larger than the perturbation along $\mathcal{S}$ that reaches the region $\mathcal{R}_k$, as $\mathcal{B} \subseteq \mathcal{R}_k$. Therefore, any upper bound derived when the boundary is the sphere of radius $1/\kappa$; i.e., $\mathscr{B}_k = \partial\mathcal{B}$ is also a valid upper bound for boundary $\mathscr{B}_k$ (see Fig. B.2 (a)). It is therefore sufficient to derive an upper bound in the worst case scenario where the boundary $\mathscr{B}_k = \partial\mathcal{B}$, and we consider this case for the remainder of the proof of the upper bound.

We now consider the linear classifier whose boundary is tangent to $\mathscr{B}_k$ at $\boldsymbol{x}^*$. For a randomly chosen subspace $\mathcal{S}$, we denote by $\boldsymbol{r}^{\mathcal{T}}_{\mathcal{S}}$ the worst-case subspace perturbation for this linear classifier. We then focus on the intersection between the boundary $\mathscr{B}_k$ and the two-dimensional plane $\mathcal{U}$ spanned by the vectors $\boldsymbol{r}^k$ and $\boldsymbol{r}^{\mathcal{T}}_{\mathcal{S}}$. This *normal* section of the boundary cuts the ball $\mathcal{B}$ through its center as the tangent spaces of the decision

[1] For a fixed point $\boldsymbol{x}^*$ on the boundary, the maximal radius $1/\kappa$ might not be achieved. To prove the result in the general case where the supremum is not achieved, one can consider instead a sequence $(\kappa_n)_n$ converging to $\kappa$, such that the balls of radius $1/\kappa_n$ and intersecting the boundary at $\boldsymbol{x}^*$ are included in $\mathcal{R}_k$. The same proof and results follow by taking the limit on the bounds derived with ball of radius $1/\kappa_n$.

boundary and the ball coincide. See Figure B.2 for a clarifying figure of this two-dimensional cross-section. We define the angle $\hat{\theta}$ as denoted in Figure B.2, such that $\cos(\hat{\theta}) = \frac{\|\boldsymbol{r}^k\|_2}{\|\boldsymbol{r}^{\mathcal{T}}_{\mathcal{S}}\|_2}$.

We apply our result on linear classifiers in Theorem 3 for the tangent classifier. We have

$$\frac{1}{\cos(\hat{\theta})^2} = \frac{\|\boldsymbol{r}^{\mathcal{T}}_{\mathcal{S}}\|_2^2}{\|\boldsymbol{r}^k\|_2^2} \leq \frac{1}{\alpha^2}\zeta_2(m,\delta), \tag{B.40}$$

with probability exceeding $1-2\delta$. Hence, using $\tan^2(\hat{\theta}) \leq (\cos^2(\hat{\theta}))^{-1}$ and the assumption of the theorem, we deduce that

$$\tan^2(\hat{\theta}) \leq \frac{1}{\alpha^2}\zeta_2(m,\delta) \leq \frac{0.2}{\kappa\|\boldsymbol{r}^k\|_2},$$

with probability exceeding $1-2\delta$. Note moreover that

$$\|\boldsymbol{r}^k\|_2\kappa \leq \frac{0.2\alpha^2}{\zeta_2(m,\delta)} < 1.$$

Hence, the assumptions of Lemma 8 hold with probability larger than $1-2\delta$. Using the notations of Figure B.2, we therefore obtain from Lemma 8

$$\frac{\|\boldsymbol{x}_\gamma - \boldsymbol{x}_0\|_2}{\|\boldsymbol{r}^{\mathcal{T}}_{\mathcal{S}}\|_2} - 1 \leq C_2\kappa\|\boldsymbol{r}^k\|_2\tan^2(\hat{\theta}) \tag{B.41}$$

with probability larger than $1-2\delta$.

Observe that $\|\boldsymbol{x}_\gamma - \boldsymbol{x}_0\|_2 \geq \|\boldsymbol{r}^k_{\mathcal{S}}\|_2$, and that $\tan^2(\hat{\theta}) \leq \frac{\|\boldsymbol{r}^T_{\mathcal{S}}\|_2^2}{\|\boldsymbol{r}^k\|_2^2}$. Hence, we obtain by re-writing Eq. (B.41)

$$\mathbb{P}\left(\frac{\|\boldsymbol{r}^k_{\mathcal{S}}\|_2^2}{\|\boldsymbol{r}^k\|_2^2} \leq \left\{1 + C_2\kappa\|\boldsymbol{r}^k\|_2\frac{\|\boldsymbol{r}^{\mathcal{T}}_{\mathcal{S}}\|_2^2}{\|\boldsymbol{r}^k\|_2^2}\right\}^2 \frac{\|\boldsymbol{r}^{\mathcal{T}}_{\mathcal{S}}\|_2^2}{\|\boldsymbol{r}^k\|_2^2}\right) \geq 1-2\delta. \tag{B.42}$$

Using the inequality in Eq. (B.40), we obtain

$$\mathbb{P}\left(\frac{\|\boldsymbol{r}^k_{\mathcal{S}}\|_2^2}{\|\boldsymbol{r}^k\|_2^2} \leq \left\{1 + C_2\kappa\|\boldsymbol{r}^k\|_2\frac{\zeta_2(m,\delta)}{\alpha^2}\right\}^2 \frac{\zeta_2(m,\delta)}{\alpha^2}\right) \geq 1-2\delta,$$

which concludes the proof of the upper bound. □

*Proof of the lower bound.* We now consider the ball $\mathcal{B}'$ of center $\boldsymbol{z}^*$ and radius $1/\kappa = \|\boldsymbol{z}^* - \boldsymbol{x}^*\|_2$ that is included in the region $\mathcal{R}_{\hat{k}(\boldsymbol{x}_0)}$. Since the ball $\mathcal{B}'$ is, by definition, included in the region $\mathcal{R}_{\hat{k}(\boldsymbol{x}_0)}$, any lower bound on $\|\boldsymbol{r}^k_{\mathcal{S}}\|_2$ when the decision boundary coincides with $\partial\mathcal{B}'$ is also a valid lower bound for any $\mathscr{B}_k$ (see Fig. B.3 (a)). We consider this case in the remainder of the proof.

To derive the lower bound, we consider the cross-section $\mathcal{U}'$ spanned by the vectors $\boldsymbol{r}^k_{\mathcal{S}}$ and $\boldsymbol{r}^k$ (Figure B.3 (b)). We have $\|\boldsymbol{r}^k\|_2\kappa < 1$; using the lower bound of Lemma 8, we obtain

$$-C_1\kappa\|\boldsymbol{r}^k\|_2\tan^2(\tilde{\theta}) \leq \frac{\|\boldsymbol{r}^k_{\mathcal{S}}\|_2}{\|\boldsymbol{x}_{\mathcal{T}} - \boldsymbol{x}_0\|_2} - 1 \tag{B.43}$$

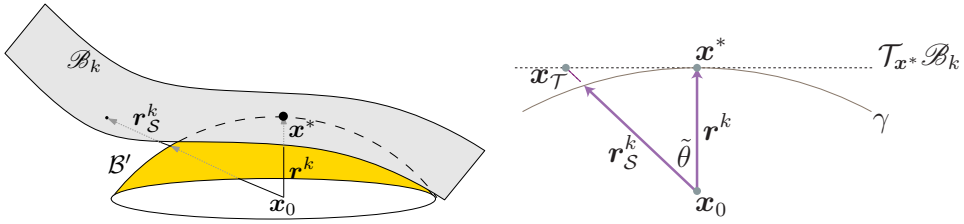



Figure B.3: Left: To prove the lower bound, we consider a ball $\mathcal{B}'$ included in $\mathcal{R}_{\hat{k}(\boldsymbol{x}_0)}$ that intersects with the boundary at $\boldsymbol{x}^*$. Lower bounds on $\|\boldsymbol{r}_{\mathcal{S}}^k\|_2$ derived when the boundary is the sphere $\partial\mathcal{B}'$ are also valid lower bounds for the real boundary $\mathscr{B}_k$. Right: Cross section of the problem along the plane $\mathcal{U}' = \text{span}\left(\boldsymbol{r}_{\mathcal{S}}^k, \boldsymbol{r}^k\right)$. $\gamma$ denotes the normal section of $\mathscr{B}_k = \mathcal{B}'$ along the plane $\mathcal{U}'$.

for any $\mathcal{S}$. Observe moreover that

$$\tan^2(\tilde{\theta}) \leq \frac{1}{\cos(\tilde{\theta})^2} = \frac{\|\boldsymbol{x}_{\mathcal{T}} - \boldsymbol{x}_0\|_2^2}{\|\boldsymbol{r}^k\|_2^2}.$$

Hence, the following bound holds:

$$\frac{\|\boldsymbol{x}_{\mathcal{T}} - \boldsymbol{x}_0\|_2^2}{\|\boldsymbol{r}^k\|_2^2}\left(1 - C_1\kappa\|\boldsymbol{r}^k\|_2\frac{\|\boldsymbol{x}_{\mathcal{T}} - \boldsymbol{x}_0\|_2^2}{\|\boldsymbol{r}^k\|_2^2}\right)^2 \leq \frac{\|\boldsymbol{r}_{\mathcal{S}}^k\|_2^2}{\|\boldsymbol{r}^k\|_2^2}.$$

Let $\boldsymbol{r}_{\mathcal{S}}^{\mathcal{T}}$ denote the worst-case perturbation belonging to subspace $\mathcal{S}$ for the *linear* classifier $\mathcal{T}_{x^*}\mathscr{B}_k$. It is not hard to see that $\boldsymbol{r}_{\mathcal{S}}^{\mathcal{T}}$ is *collinear* to $\boldsymbol{r}_{\mathcal{S}}^k$ (see Lemma 10 for a proof). Hence, we have $\boldsymbol{r}_{\mathcal{S}}^{\mathcal{T}} = \boldsymbol{x}_{\mathcal{T}} - \boldsymbol{x}_0$. By applying our result on linear classifiers in Theorem 3 for the tangent classifier $\mathcal{T}_{x^*}\mathscr{B}_k$, we have:

$$\mathbb{P}\left(\frac{\zeta_1(m,\delta)}{\alpha^2} \leq \frac{\|\boldsymbol{r}_{\mathcal{S}}^{\mathcal{T}}\|_2^2}{\|\boldsymbol{r}^k\|_2^2} \leq \frac{\zeta_2(m,\delta)}{\alpha^2}\right) \geq 1 - 2\delta.$$

We therefore conclude that

$$\mathbb{P}\left(\frac{\zeta_1(m,\delta)}{\alpha^2}\left\{1 - C_1\kappa\|\boldsymbol{r}^k\|_2\frac{\zeta_2(m,\delta)}{\alpha^2}\right\}^2 \leq \frac{\|\boldsymbol{r}_{\mathcal{S}}^k\|_2^2}{\|\boldsymbol{r}^k\|_2^2}\right) \geq 1 - 2\delta,$$

which concludes the proof of the lower bound.

□

The goal is now to extend the previous result, derived for binary classifiers, to the multiclass classification case. To do so, we show the following lemma.

**Lemma 9** (Binary case to multiclass)**.** *Let* $p = \arg\min_i \|\boldsymbol{r}^i\|_2$*. Define the deterministic set*

$$A = \left\{k : \|\boldsymbol{r}^k\|_2 \geq 1.45\sqrt{\zeta_2(m,\delta)}\sqrt{\frac{d}{m}}\|\boldsymbol{r}^*\|_2\right\}. \tag{B.44}$$

*Assume that, for all $k \in A^c$, we have*

$$\mathbb{P}\left(l \leq \frac{\|\boldsymbol{r}_{\mathcal{S}}^k\|_2}{\|\boldsymbol{r}^k\|_2} \leq u\right) \geq 1-\delta. \tag{B.45}$$

*and that*

$$\mathbb{P}\left(\|\boldsymbol{r}_{\mathcal{S}}^p\|_2 \geq 1.45\sqrt{\zeta_2(m,\delta)}\sqrt{\frac{d}{m}}\|\boldsymbol{r}^*\|_2\right) \leq t. \tag{B.46}$$

*Then, we have*

$$\mathbb{P}\left(l \leq \frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq u\right) \geq 1-(L+1)\delta - t. \tag{B.47}$$

*Proof.* Note first that

$$\mathbb{P}\left(\frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \geq u\right) \leq \mathbb{P}\left(\left\{\frac{\|\boldsymbol{r}_{\mathcal{S}}^p\|_2}{\|\boldsymbol{r}^p\|_2} \geq u\right\}\right) \leq \delta. \tag{B.48}$$

We now focus on bounding the other bad event probability $\mathbb{P}\left(\frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq l\right)$. We have

$$\mathbb{P}\left(\frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq l\right) = \mathbb{P}\left(\min_{k\notin A}\|\boldsymbol{r}_{\mathcal{S}}^k\|_2 = \|\boldsymbol{r}_{\mathcal{S}}^*\|_2, \frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq l\right) + \mathbb{P}\left(\min_{k\in A}\|\boldsymbol{r}_{\mathcal{S}}^k\|_2 = \|\boldsymbol{r}_{\mathcal{S}}^*\|_2, \frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq l\right) \tag{B.49}$$

The first probability can be bounded as follows:

$$\mathbb{P}\left(\min_{k\notin A}\|\boldsymbol{r}_{\mathcal{S}}^k\|_2 = \|\boldsymbol{r}_{\mathcal{S}}^*\|_2, \frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq l\right) \leq \mathbb{P}\left(\bigcup_{k\notin A}\frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq l\right) \leq L\delta. \tag{B.50}$$

The second probability can also be bounded in the following way

$$\mathbb{P}\left(\min_{k\in A}\|\boldsymbol{r}_{\mathcal{S}}^k\|_2 = \|\boldsymbol{r}_{\mathcal{S}}^*\|_2, \frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq l\right) \leq \mathbb{P}\left(\min_{k\in A}\|\boldsymbol{r}_{\mathcal{S}}^k\|_2 = \|\boldsymbol{r}_{\mathcal{S}}^*\|_2\right) = \mathbb{P}\left(\exists k \in A, \|\boldsymbol{r}_{\mathcal{S}}^k\|_2 \leq \|\boldsymbol{r}_{\mathcal{S}}^*\|_2\right). \tag{B.51}$$

Observe that, for $k \in A$, we have $\|\boldsymbol{r}_{\mathcal{S}}^k\|_2 \geq \|\boldsymbol{r}^k\|_2 \geq 1.45\sqrt{\zeta_2(m,\delta)}\sqrt{\frac{d}{m}}\|\boldsymbol{r}^*\|_2$. Hence, we conclude that

$$\mathbb{P}\left(\min_{k\in A}\|\boldsymbol{r}_{\mathcal{S}}^k\|_2 = \|\boldsymbol{r}_{\mathcal{S}}^*\|_2, \frac{\|\boldsymbol{r}_{\mathcal{S}}^*\|_2}{\|\boldsymbol{r}^*\|_2} \leq l\right) \leq \mathbb{P}\left(1.45\sqrt{\zeta_2(m,\delta)}\sqrt{\frac{d}{m}}\|\boldsymbol{r}^*\|_2 \leq \|\boldsymbol{r}_{\mathcal{S}}^*\|_2\right) \tag{B.52}$$

$$\leq \mathbb{P}\left(1.45\sqrt{\zeta_2(m,\delta)}\sqrt{\frac{d}{m}}\|\boldsymbol{r}^*\|_2 \leq \|\boldsymbol{r}_{\mathcal{S}}^p\|_2\right) \leq t. \tag{B.53}$$

□

**Corollary 1.** *Let $\mathcal{S}$ be a random $m$-dimensional subspace of $\mathbb{R}^d$. Assume that, for all $k \notin A$, we have*

$$\kappa(\mathscr{B}_k)\|\boldsymbol{r}^k\|_2 \leq \frac{0.2}{\zeta_2(m,\delta)}\frac{m}{d} \tag{B.54}$$

*Then, we have*

$$0.875\sqrt{\zeta_1(m,\delta)}\sqrt{\frac{d}{m}} \leq \frac{\|\boldsymbol{r}^*_{\mathcal{S}}\|_2}{\|\boldsymbol{r}^*\|_2} \leq 1.45\sqrt{\zeta_2(m,\delta)}\sqrt{\frac{d}{m}} \tag{B.55}$$

*with probability larger than $1-4(L+2)\delta$.*

*Proof.* Using Theorem 4, we have that for all $k \notin A$, the result in Eq. (B.39) holds with probability exceeding $1-4\delta$. We simplify the result with the assumption $\kappa(\mathscr{B}_k)\|\boldsymbol{r}^k\|_2 \leq \frac{0.2}{\zeta_2(m,\delta)}\frac{m}{d}$. Hence, the bounds of Theorem 4 can be written as follows

$$\frac{\zeta_1(m,\delta)}{\alpha^2}\left(1-0.2C_1\right)^2 \leq \frac{\|\boldsymbol{r}^k_{\mathcal{S}}\|_2^2}{\|\boldsymbol{r}^k\|_2^2} \leq \frac{\zeta_2(m,\delta)}{\alpha^2}\left(1+0.2C_2\right)^2, \tag{B.56}$$

which leads to the following bounds:

$$\zeta_1(m,\delta)\frac{d}{m}0.875^2 \leq \frac{\|\boldsymbol{r}^k_{\mathcal{S}}\|_2^2}{\|\boldsymbol{r}^k\|_2^2} \leq \zeta_2(m,\delta)\frac{d}{m}1.45^2, \tag{B.57}$$

with probability exceeding $1-4\delta$.

By using Lemma 9, together with the fact that $t=\delta$, we obtain

$$\mathbb{P}\left(0.875\sqrt{\zeta_1(m,\delta)}\sqrt{\frac{d}{m}} \leq \frac{\|\boldsymbol{r}^*_{\mathcal{S}}\|_2}{\|\boldsymbol{r}^*\|_2} \leq 1.45\sqrt{\zeta_2(m,\delta)}\sqrt{\frac{d}{m}}\right) \geq 1-4(L+2)\delta, \tag{B.58}$$

which concludes the proof. □

### B.2.1 Useful results

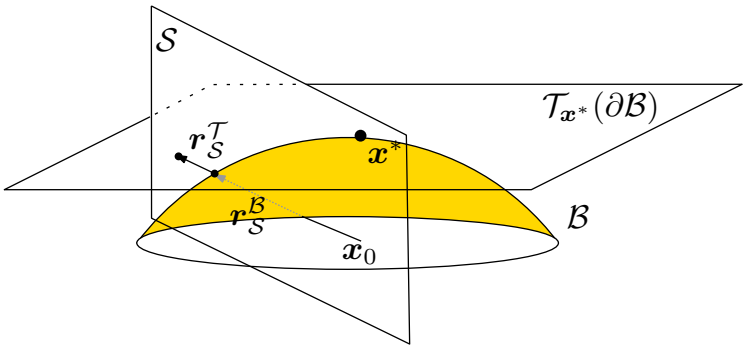



Figure B.4: The worst-case perturbation in the subspace $\mathcal{S}$ when the decision boundary is $\partial\mathcal{B}$ and $T_{\boldsymbol{x}^*}(\partial\mathcal{B})$ (denoted respectively by $\boldsymbol{r}^{\mathcal{B}}_{\mathcal{S}}$ and $\boldsymbol{r}^{\mathcal{T}}_{\mathcal{S}}$) are collinear.

**Lemma 10.** *Let $\boldsymbol{x}_0 \in \mathbb{R}^d$, and $\boldsymbol{x}^*$ denote the closest point to $\boldsymbol{x}_0$ on the sphere $\partial\mathcal{B}$ (see Fig. B.4). Let $\mathcal{T}_{\boldsymbol{x}^*}(\partial\mathcal{B})$ be the tangent space to $\partial\mathcal{B}$ at $\boldsymbol{x}^*$. For an arbitrary subspace $\mathcal{S}$, let*

*$\boldsymbol{r}_{\mathcal{S}}^{\mathcal{T}}$ and $\boldsymbol{r}_{\mathcal{S}}^{\mathcal{B}}$ denote the worst-case perturbations of $\boldsymbol{x}_0$ on the subspace $\mathcal{S}$, when the decision boundaries are respectively $\mathcal{T}_{\boldsymbol{x}^*}(\partial\mathcal{B})$ and $\partial\mathcal{B}$. Then, the two perturbations $\boldsymbol{r}_{\mathcal{S}}^{\mathcal{T}}$ and $\boldsymbol{r}_{\mathcal{S}}^{\mathcal{B}}$ are collinear.*

*Proof.* Assuming the center of the ball $\mathcal{B}$ is the origin, the points on the sphere $\partial\mathcal{B}$ satisfy equation: $\|\boldsymbol{x}\|_2 = R$, where $R$ denotes the radius. Hence, the perturbation $\boldsymbol{r}_{\mathcal{S}}^{\mathcal{B}}$ is given by

$$\boldsymbol{r}_{\mathcal{S}}^{\mathcal{B}} = \underset{\boldsymbol{r}\in\mathbb{R}^d}{\operatorname{argmin}} \|\boldsymbol{r}\|_2^2 \text{ such that } \|\boldsymbol{x}_0 + \mathbf{P}_{\mathcal{S}}\boldsymbol{r}\|_2^2 = R^2. \tag{B.59}$$

By equating the gradient of Lagrangian of the above constrained optimization problem to zero, we obtain the following necessary optimality condition

$$\boldsymbol{r} + \lambda\mathbf{P}_{\mathcal{S}}(\boldsymbol{x}_0 + \mathbf{P}_{\mathcal{S}}\boldsymbol{r}) = 0.$$

It should further be noted that $\mathbf{P}_{\mathcal{S}}\boldsymbol{r}_{\mathcal{S}}^{\mathcal{B}} = \boldsymbol{r}_{\mathcal{S}}^{\mathcal{B}}$. Indeed, if $\boldsymbol{r}_{\mathcal{S}}^{\mathcal{B}}$ had a component orthogonal to $\mathcal{S}$, the projection of $\boldsymbol{r}_{\mathcal{S}}^{\mathcal{B}}$ onto $\mathcal{S}$ would have strictly smaller $\ell_2$ norm, while still satisfying the condition in Eq.(B.59). Hence, the necessary condition of optimality becomes

$$(1+\lambda)\boldsymbol{r} + \lambda\mathbf{P}_{\mathcal{S}}\boldsymbol{x}_0 = 0,$$

from which we conclude that $\boldsymbol{r}_{\mathcal{S}}^{\mathcal{B}}$ is collinear to $\mathbf{P}_{\mathcal{S}}\boldsymbol{x}_0$.

It should further be noted that $\boldsymbol{r}_{\mathcal{S}}^{\mathcal{T}}$ can be computed in closed form (see Fact 5), and is collinear to $\mathbf{P}_{\mathcal{S}}(\boldsymbol{x}^* - \boldsymbol{x}_0)$, which is itself collinear to $\boldsymbol{x}_0$, as the the center of the ball was assumed to be the origin. This concludes the proof. □

**Lemma 11.** *If $x \in [0, 2(\sqrt{2}-1)]$,*

$$\sqrt{1-x} \geq 1 - \frac{x}{2} - \frac{x^2}{4}. \tag{B.60}$$

**Lemma 12.** *If $x \geq 0$,*

$$\sqrt{1+x} \geq 1 + \frac{x}{2} - \frac{x^2}{8}. \tag{B.61}$$

# Bibliography


[ABB15] S. An, F. Boussaid, and M. Bennamoun. "How Can Deep Rectifier Networks Achieve Linear Separability and Preserve Distances?" In: *International Conference on Machine Learning (ICML)*. 2015, pp. 514–523.

[ALM15] S. Arora, Y. Liang, and T. Ma. "Why are deep nets reversible: A simple theory, with implications for training". In: *arXiv preprint arXiv:1511.05653* (2015).

[ARP16] F. Anselmi, L. Rosasco, and T. Poggio. "On invariance and selectivity in representation learning". In: *Information and Inference* (2016).

[Ash07] J. Ashburner. "A fast diffeomorphic image registration algorithm". In: *Neuroimage* 38.1 (2007), pp. 95–113.

[Bak+16] A. Bakry, M. Elhoseiny, T. El-Gaaly, and A. Elgammal. "Digging Deep into the layers of CNNs: In Search of How CNNs Achieve View Invariance". In: *International Conference on Learning Representations (ICLR)*. 2016.

[Bar+06] M. Barreno, B. Nelson, R. Sears, A. Joseph, and D. Tygar. "Can machine learning be secure?" In: *ACM Symposium on Information, computer and communications security*. 2006, pp. 16–25.

[Bay+08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. "Speeded-up robust features (SURF)". In: *Computer vision and image understanding* 110.3 (2008), pp. 346–359.

[BC11] F. Benmansour and L. D. Cohen. "Tubular structure segmentation based on minimal path method and anisotropic enhancement". In: *International Journal of Computer Vision* 92.2 (2011), pp. 192–210.

[BCV13] Y. Bengio, A. Courville, and P. Vincent. "Representation learning: A review and new perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828.

[BE02] O. Bousquet and A. Elisseeff. "Stability and generalization". In: *The Journal of Machine Learning Research* 2 (2002), pp. 499–526.

[Bha04] C. Bhattacharyya. "Robust classification of noisy data using second order cone programming approach". In: *Intelligent Sensing and Information Processing*. 2004, pp. 433–438.

[Big+13] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli. "Evasion attacks against machine learning at test time". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 2013, pp. 387–402.

[BM13] J. Bruna and S. Mallat. "Invariant scattering convolution networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1872–1886.

[BNL12] B. Biggio, B. Nelson, and P. Laskov. "Poisoning attacks against support vector machines". In: *International Conference on Machine Learning (ICML)*. 2012.

[BPL10] Y.-L. Boureau, J. Ponce, and Y. LeCun. "A theoretical analysis of feature pooling in visual recognition". In: *International Conference on Machine Learning (ICML)*. 2010, pp. 111–118.

[BSL14] J. Bruna, A. Szlam, and Y. LeCun. "Signal recovery from pooling representations". In: *International Conference on Machine Learning (ICML)*. 2014.

[Car+16] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. "Hidden Voice Commands". In: *25th USENIX Security Symposium (USENIX Security 16), Austin, TX*. 2016.

[Cha+10] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin. "Training and testing low-degree polynomial data mappings via linear SVM". In: *The Journal of Machine Learning Research* 11 (2010), pp. 1471–1490.

[Cha+14] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. "Return of the Devil in the Details: Delving Deep into Convolutional Nets". In: *British Machine Vision Conference*. 2014.

[Cho+14] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. "The loss surfaces of multilayer networks". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2014.

[CJF16] O. Canévet, C. Jose, and F. Fleuret. "Importance Sampling Tree for Large-scale Empirical Expectation". In: *International Conference on Machine Learning (ICML)*. 2016, pp. 1454–1462.

[CL11] C.-C. Chang and C.-J. Lin. "LIBSVM: a library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011), p. 27.

[CL83] M. G. Crandall and P.-L. Lions. "Viscosity solutions of Hamilton–Jacobi equations". In: *Transactions of the American Mathematical Society* 277.1 (1983), pp. 1–42.

[CM08] C. Caramanis and S. Mannor. "Learning in the Limit with Adversarial Disturbances." In: *Interational Conference on Learning Theory (COLT)*. 2008, pp. 467–478.

[CMX12] C. Caramanis, S. Mannor, and H. Xu. "Robust optimization in machine learning". In: *Optimization for machine learning*. Ed. by S. Sra, S. Nowozin, and S. J. Wright. Mit Press, 2012. Chap. 14.

[CPE14] K. Chalupka, P. Perona, and F. Eberhardt. "Visual Causal Feature Learning". In: *arXiv preprint arXiv:1412.2309* (2014).

[Dal+04] N. Dalvi, P. Domingos, S. Sanghai, and D. Verma. "Adversarial classification". In: *ACM SIGKDD*. 2004, pp. 99–108.

[Dau+14] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization". In: *Advances in Neural Information Processing Systems (NIPS)*. 2014, pp. 2933–2941.

[DB16] A. Dosovitskiy and T. Brox. "Inverting Visual Representations with Convolutional Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[DC07] J. J. DiCarlo and D. D. Cox. "Untangling invariant object recognition". In: *Trends in cognitive sciences* 11.8 (2007), pp. 333–341.

[DG03] S. Dasgupta and A. Gupta. "An elementary proof of a theorem of Johnson and Lindenstrauss". In: *Random Structures & Algorithms* 22.1 (2003), pp. 60–65.

[DG05] D. Donoho and C. Grimes. "Image manifolds which are isometric to Euclidean space". In: *Journal of mathematical imaging and vision* 23.1 (2005), pp. 5–24.

[Dij59] E. W. Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1 (1959), pp. 269–271.

[Don+15] J. Dong, N. Karianakis, D. Davis, J. Hernandez, J. Balzer, and S. Soatto. "Multiview Feature Engineering and Learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[DS15] J. Dong and S. Soatto. "Domain Size Pooling in Local Descriptors: DSP-SIFT". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[DSX10] O. Dekel, O. Shamir, and L. Xiao. "Learning to classify with missing and corrupted features". In: *Machine learning* 81.2 (2010), pp. 149–178.

[DT05] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 2005, pp. 886–893.

[Fan+08a] R.-W. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. "LIBLINEAR: A library for large linear classification". In: *The Journal of Machine Learning Research* 9 (2008), pp. 1871–1874.

[Fan+08b] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. "LIBLINEAR: A library for large linear classification". In: *The Journal of Machine Learning Research* 9 (2008), pp. 1871–1874.

[Faw+16] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard. "Adaptive data augmentation for image classification". In: *International Conference on Image Processing (ICIP)*. 2016.

[FDF15] A. Fawzi, M. Davies, and P. Frossard. "Dictionary learning for fast classification based on soft-thresholding". In: *International Journal of Computer Vision* 114.2-3 (2015), pp. 306–321.

[FF13] A. Fawzi and P. Frossard. "Image registration with sparse approximations in parametric dictionaries". In: *SIAM Journal on Imaging Sciences* 6.4 (2013), pp. 2370–2403.

[FF15] A. Fawzi and P. Frossard. "Manitest: Are classifiers really invariant?" In: *British Machine Vision Conference (BMVC)*. 2015, pp. 106.1–106.13.

[FF16] A. Fawzi and P. Frossard. "Measuring the effect of nuisance variables on classifiers". In: *British Machine Vision Conference (BMVC)*. 2016.

[FFF15a] A. Fawzi, O. Fawzi, and P. Frossard. "Analysis of classifiers' robustness to adversarial perturbations". In: *arXiv preprint arXiv:1502.02590* (2015).

[FFF15b] A. Fawzi, O. Fawzi, and P. Frossard. "Fundamental limits on adversarial robustness". In: *Proceedings of ICML, Workshop on Deep Learning*. 2015.

[FMDF16] A. Fawzi, S. Moosavi-Dezfooli, and P. Frossard. "Robustness of classifiers: from adversarial to random noise". In: *Neural Information Processing Systems (NIPS)*. 2016.

[Fre+15] O. Freifeld, S. Hauberg, K. Batmanghelich, and J. W. F. III. "Highly-Expressive Spaces of Well-Behaved Transformations: Keeping It Simple". In: *International Conference on Computer Vision (ICCV)*. Santiago, Chile, Dec. 2015.

[GBB11] X. Glorot, A. Bordes, and Y. Bengio. "Deep sparse rectifier neural networks". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2011, pp. 315–323.

[GBS05] J. Geusebroek, G. Burghouts, and A. Smeulders. "The Amsterdam library of object images". In: *International Journal of Computer Vision* 61.1 (2005), pp. 103–112.

[GE08] Y. Goldberg and M. Elhadad. "splitSVM: fast, space-efficient, non-heuristic, polynomial kernel computation for NLP applications". In: *46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. 2008, pp. 237–240.

[Goo+09] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng. "Measuring invariances in deep networks". In: *Advances in Neural Information Processing Systems*. 2009, pp. 646–654.

[Goo15] I. Goodfellow. "Adversarial examples". Presentation at the Deep Learning Summer School, Montreal. 2015. URL: http://www.iro.umontreal.ca/~memisevr/dlss2015/goodfellow_adv.pdf.

[GR06] A. Globerson and S. Roweis. "Nightmare at test time: robust learning by feature deletion". In: *International Conference on Machine Learning (ICML)*. 2006, pp. 353–360.

[GR14] S. Gu and L. Rigazio. "Towards Deep Neural Network Architectures Robust to Adversarial Examples". In: *arXiv preprint arXiv:1412.5068* (2014).

[GSS15] I. J. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations (ICLR)*. 2015.

[Hau+16] S. Hauberg, O. Freifeld, A. Larsen, J. Fisher III, and L. Hansen. "Dreaming More Data: Class-dependent Distributions over Diffeomorphisms for Learned Data Augmentation". In: *Artificial Intelligence and Statistics (AISTATS)*. 2016.

[Hua+15] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári. "Learning with a strong adversary". In: *CoRR, abs/1511.03034* (2015).

[Jar+09] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. "What is the best multi-stage architecture for object recognition?" In: *International Conference on Computer Vision (ICCV)*. 2009, pp. 2146–2153.

[JDV08] L. Jacques and C. De Vleeschouwer. "A geometrical study of matching pursuit parametrization". In: *IEEE Transactions on Signal Processing* 56.7 (2008), pp. 2835–2848.

[Jia+14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. "Caffe: Convolutional architecture for fast feature embedding". In: *ACM International Conference on Multimedia (MM)*. 2014, pp. 675–678.

[JSZ+15] M. Jaderberg, K. Simonyan, A. Zisserman, et al. "Spatial transformer networks". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2008–2016.

[KB14] D. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *International Conference on Learning Representations (ICLR)*. 2014.

[KDS16] N. Karianakis, J. Dong, and S. Soatto. "An Empirical Evaluation of Current Convolutional Architectures' Ability to Manage Nuisance Location and Scale Variability". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[KF09] E. Kokiopoulou and P. Frossard. "Minimum distance between pattern transformation manifolds: Algorithm and applications". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.7 (2009), pp. 1225–1238.

[KH09] A. Krizhevsky and G. Hinton. "Learning multiple layers of features from tiny images". In: *Master's thesis, Department of Computer Science, University of Toronto* (2009).

[KS98] R. Kimmel and J. A. Sethian. "Computing geodesic paths on manifolds". In: *Proceedings of the National Academy of Sciences* 95.15 (1998), pp. 8431–8435.

[KSH12] A. Krizhevsky, I. Sutskever, and G. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems (NIPS)*. 2012, pp. 1106–1114.

[Lan+03] G. Lanckriet, L. Ghaoui, C. Bhattacharyya, and M. Jordan. "A robust minimax approach to classification". In: *The Journal of Machine Learning Research* 3 (2003), pp. 555–582.

[LCB07] G. Loosli, S. Canu, and L. Bottou. "Training invariant support vector machines using selective sampling". In: *Large scale kernel machines* (2007), pp. 301–320.

[LCY14] M. Lin, Q. Chen, and S. Yan. "Network In Network". In: *International Conference on Learning Representations (ICLR)*. 2014.

[LeC+98a] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[LeC+98b] Y. LeCun, L. Bottou, G. B. Orr, and K. R. Muller. "Efficient BackProp". In: *Neural Networks: Tricks of the Trade*. 1998, pp. 9–50.

[LeC+99] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio. "Object recognition with gradient-based learning". In: *Shape, contour and grouping in computer vision*. 1999, pp. 319–345.

[Lee09] J. M. Lee. *Manifolds and differential geometry*. Vol. 107. American Mathematical Society Providence, 2009.

[LHL15] C. Lyu, K. Huang, and H.-N. Liang. "A Unified Gradient Regularization Family for Adversarial Examples". In: *arXiv preprint arXiv:1511.06385* (2015).

[Lin03] Q. Lin. "Enhancement, extraction, and visualization of 3D volume data". In: *PhD thesis* (2003).

[LL94] X. Luo and Z. Luo. "Extension of Hoffman's error bound to polynomial systems". In: *SIAM Journal on Optimization* 4.2 (1994), pp. 383–392.

[LMP14] G Li, B. Mordukhovich, and T. Pham. "New fractional error bounds for polynomial systems with applications to Hölderian stability in optimization and spectral theory of tensors". In: *Mathematical Programming* (2014), pp. 1–30.

[Low04] D. Lowe. "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[LP94a] G. Lugosi and M. Pawlak. "On the posterior-probability estimate of the error rate of nonparametric classification rules". In: *IEEE Transactions on Information Theory* 40.2 (1994), pp. 475–481.

[LP94b] Z.-Q. Luo and J.-S. Pang. "Error bounds for analytic systems and their applications". In: *Mathematical Programming* 67.1-3 (1994), pp. 1–28.

[LP98] A. Lewis and J. Pang. "Error bounds for convex inequality systems". In: *Generalized convexity, generalized monotonicity: recent results*. Springer, 1998, pp. 75–110.

[Luo+15] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao. "Foveation-based Mechanisms Alleviate Adversarial Examples". In: *arXiv preprint arXiv:1511.06292* (2015).

[LV15] K. Lenc and A. Vedaldi. "Understanding image representations by measuring their equivariance and equivalence". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 991–999.

[Mae+97] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. "Multimodality image registration by maximization of mutual information". In: *IEEE transactions on Medical Imaging* 16.2 (1997), pp. 187–198.

[Mal12] S. Mallat. "Group invariant scattering". In: *Communications on Pure and Applied Mathematics* 65.10 (2012), pp. 1331–1398.

[Mar10] J. Martens. "Deep learning via Hessian-free optimization". In: *International Conference on Machine Learning (ICML)*. 2010, pp. 735–742.

[MDFF16] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. "DeepFool: a simple and accurate method to fool deep neural networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[Met+53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. "Equation of state calculations by fast computing machines". In: *The Journal Of Chemical Physics* 21.6 (1953), pp. 1087–1092.

[MG15] J. Martens and R. Grosse. "Optimizing neural networks with Kronecker-factored approximate curvature". In: *arXiv preprint arXiv:1503.05671* (2015).

[MHN13] A. L. Maas, A. Y. Hannun, and A. Y. Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *International Conference on Machine Learning (ICML)*. 2013.

[Mir14] J.-M. Mirebeau. "Anisotropic fast-marching on cartesian grids using lattice basis reduction". In: *SIAM Journal on Numerical Analysis* 52.4 (2014), pp. 1573–1599.

[Mon+14] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio. "On the number of linear regions of deep neural networks". In: *Advances In Neural Information Processing Systems*. 2014, pp. 2924–2932.

[MS05] K. Mikolajczyk and C. Schmid. "A performance evaluation of local descriptors". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.10 (2005), pp. 1615–1630.

[MV15] A. Mahendran and A. Vedaldi. "Understanding deep image representations by inverting them". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5188–5196.

[NZ03] K. Ng and X. Zheng. "Error bounds of constrained quadratic functions and piecewise affine inequality systems". In: *Journal Of Optimization Theory And Applications* 118.3 (2003), pp. 601–618.

[OM15] E. Oyallon and S. Mallat. "Deep roto-translation scattering for object classification". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 2865–2873.

[Osa+16] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, and D. Perez-Cabo. *No Bot Expects the DeepCAPTCHA! Introducing Immutable Adversarial Examples with Applications to CAPTCHA*. Cryptology ePrint Archive, Report 2016/336. http://eprint.iacr.org/2016/336. 2016.

[Pan97] J. Pang. "Error bounds in mathematical programming". In: *Mathematical Programming* 79.1-3 (1997), pp. 299–332.

[Pap+15] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. "Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks". In: *arXiv preprint arXiv:1511.04508* (2015).

[Pau+14] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, and C. Schmid. "Transformation pursuit for image classification". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 3646–3653.

[Pen+10] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010, pp. 763–770.

[Pey+10] G. Peyré, M. Péchaud, R. Keriven, and L. D. Cohen. "Geodesic methods in computer vision and graphics". In: *Foundations and Trends in Computer Graphics and Vision* 5.3–4 (2010), pp. 197–397.

[PVZ15] O. M. Parkhi, A. Vedaldi, and A. Zisserman. "Deep face recognition". In: *British Machine Vision Conference (BMVC)* 1.3 (2015), p. 6.

[Rus+15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. "Imagenet large scale visual recognition challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[Rus06] A. P. Ruszczyński. *Nonlinear optimization*. Vol. 13. Princeton university press, 2006.

[SC16] S. Soatto and A. Chiuso. "Visual Representations: Defining properties and deep approximation". In: *International Conference on Learning Representations (ICLR)* (2016).

[SDK15] S. Soatto, J. Dong, and N. Karianakis. "Visual Scene Representations: Contrast, Scaling and Occlusion". In: *International Conference on Learning Representations (ICLR) Workshop*. 2015.

[Sim+00] P. Simard, Y. Le Cun, J. Denker, and B. Victorri. "Transformation invariance in pattern recognition: Tangent distance and propagation". In: *International Journal of Imaging Systems and Technology* 11.3 (2000), pp. 181–197.

[SM13] L. Sifre and S. Mallat. "Rotation, scaling and deformation invariant scattering for texture discrimination". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 1233–1240.

[SV00] J. A. Sethian and A. Vladimirsky. "Fast methods for the Eikonal and related Hamilton–Jacobi equations on unstructured meshes". In: *Proceedings of the National Academy of Sciences* 97.11 (2000), pp. 5699–5703.

[SV03] J. A. Sethian and A. Vladimirsky. "Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms". In: *SIAM Journal on Numerical Analysis* 41.1 (2003), pp. 325–363.

[SVZ13] K. Simonyan, A. Vedaldi, and A. Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013).

[SYN15] U. Shaham, Y. Yamada, and S. Negahban. "Understanding Adversarial Training: Increasing Local Stability of Neural Nets through Robust Optimization". In: *arXiv preprint arXiv:1511.05432* (2015).

[SZ14] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *International Conference on Learning Representations (ICLR)*. 2014.

[Sze+14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. "Intriguing properties of neural networks". In: *International Conference on Learning Representations (ICLR)*. 2014.

[Sze+15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[Sze10] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

[Tai+14] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. "Deepface: Closing the gap to human-level performance in face verification". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 1701–1708.

[TDSL00] J. Tenenbaum, V. De Silva, and J. Langford. "A global geometric framework for nonlinear dimensionality reduction". In: *Science* 290.5500 (2000), pp. 2319–2323.

[TFT01] S. J. Thorpe and M. Fabre-Thorpe. "Seeking categories in the brain". In: *Science* 291.5502 (2001), pp. 260–263.

[TG07] T. B. Trafalis and R. C. Gilbert. "Robust support vector machines for classification and computational issues". In: *Optimisation Methods and Software* 22.1 (2007), pp. 187–198.

[Tsi95] J. N. Tsitsiklis. "Efficient algorithms for globally optimal trajectories". In: *IEEE Transactions on Automatic Control* 40.9 (1995), pp. 1528–1538.

[TV16] P. Tabacof and E. Valle. "Exploring the Space of Adversarial Images". In: *IEEE International Joint Conference on Neural Networks* (2016).

[VL05] N. Vasconcelos and A. Lippman. "A multiresolution manifold distance for invariant image similarity". In: *IEEE Transactions on Multimedia* 7.1 (2005), pp. 127–142.

[VL15] A. Vedaldi and K. Lenc. "MatConvNet: Convolutional neural networks for matlab". In: *ACM International Conference on Multimedia (MM)*. 2015, pp. 689–692.

[Wah+11] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. "The Caltech-UCSD birds-200-2011 dataset". In: (2011).

[Wak+05] M. B. Wakin, D. L. Donoho, H. Choi, and R. G. Baraniuk. "The multiscale structure of non-differentiable image manifolds". In: *Wavelets XI in SPIE International Symposium on Optical Science and Technology*. 2005.

[WW90] H. F. Walker and L. T. Watson. "Least-change secant update methods for underdetermined systems". In: *SIAM Journal on numerical analysis* 27.5 (1990), pp. 1227–1262.

[XCM09] H. Xu, C. Caramanis, and S. Mannor. "Robustness and regularization of support vector machines". In: *The Journal of Machine Learning Research* 10 (2009), pp. 1485–1510.

[Xia+15] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli. "Is feature selection secure against training data poisoning?" In: *International Conference on Machine Learning (ICML)*. Vol. 37. 2015, pp. 1689–1698.

[ZF03] B. Zitova and J. Flusser. "Image registration methods: a survey". In: *Image and Vision Computing* 21.11 (2003), pp. 977–1000.

[ZF14] M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks". In: *European Conference on Computer Vision (ECCV)*. 2014, pp. 818–833.

[ZG16] Q. Zhao and L. D. Griffin. "Suppressing the Unusual: towards Robust CNNs using Symmetric Activation Functions". In: *arXiv preprint arXiv:1603.05145* (2016).

[Łoj59] S. Łojasiewicz. "Sur le probleme de la division". In: *Studia Mathematica* 18.1 (1959), pp. 87–136.

## Alhussein Fawzi


CONTACT INFORMATION

*E-mail:* alhussein.fawzi@epfl.ch
*Mobile:* +41 78 83 070 93
*WWW:* www.alhusseinfawzi.info


RESEARCH INTERESTS

Sparse coding and dictionary learning, computer vision, machine learning, signal and image processing.

EDUCATION

**Ecole Polytechnique Fédérale de Lausanne** – Lausanne, Switzerland *May 2012 - present*
*PhD in Electrical Engineering*

- Research in dictionary learning techniques for classification; invariance of classification to geometric and adversarial transformations.
- Supervisor: Prof. Pascal Frossard.

**Ecole Polytechnique Fédérale de Lausanne** – Lausanne, Switzerland *Sept 2010 - Feb 2012*
*M.Sc. in Electronics and Electrical Engineering (Information Technologies orientation)*

- Dissertation title: *Geometric group sparsity in image analysis.*
- Awarded best Master's student in Electronics and Electrical Engineering.

**Ecole Centrale de Nantes** – Nantes, France *Sept 2008 - Aug 2010*
*Diplôme d'Ingénieur (equivalent to B.Sc).*

**Classes préparatoires aux Grandes Ecoles (CPGE), Lycée Chaptal** – Paris, France *2006 - 2008*

- Concentration in Mathematics and Physics (courses include Abstract Algebra, General Topology, Real Analysis, Differential Calculus).
- The CPGE are two years of intensive university-level preparation in Mathematics and Physics for the highly selective entrance exam to the French Grandes Ecoles.

**Baccalauréat at Lycée Français du Caire** – Cairo, Egypt *2006*

- High school diploma with major in science (Mathematics, Physics and Biology) and emphasis in Mathematics.
- Diploma awarded with highest honors.

AWARDS

- Recipient twice of the IBM PhD Fellowship award (Academic years 2013-2014 and 2015-2016).
- SIA Vaudoise prize for student excellence (October 2012).
- Anna Barbara Reinhard Prize for student excellence from the Institution of Engineering and Technology (IET) (October 2012).
- Gold medal in Egyptian Olympiad in Informatics (EOI) 2004.

PUBLICATIONS

**Pre-prints**

▷ S-M. Moosavi-Dezfooli*, **A. Fawzi***, O. Fawzi, P. Frossard, *Universal adversarial perturbations*, arXiv pre-print arXiv:1610.08401, 2016. (*: Equal contribution).
▷ **A. Fawzi**, O. Fawzi, P. Frossard, *Analysis of classifiers' robustness to adversarial perturbations*, submitted to Machine Learning Journal.

**Journal papers**

▷ **A. Fawzi**, M. Sinn, P. Frossard, *Multi-task additive models with shared transfer functions*, IEEE Transactions on Signal Processing, 2016.
▷ **A. Fawzi**, J-B. Fiot, B. Chen, M. Sinn, P. Frossard, *Structured Dimensionality Reduction for Additive Model Regression*, IEEE Transactions on Data Knowledge and Engineering (TKDE), 2016.
▷ **A. Fawzi**, M. Davies, P. Frossard, *Dictionary learning for fast classification based on soft-thresholding*, International Journal of Computer Vision (IJCV), 114(2-3), pp.306-321, 2015.
▷ **A. Fawzi**, P. Frossard, *Image registration with sparse approximations in parametric dictionaries*, SIAM J. on Imaging Sci., 6(4), pp. 2370-2403, 2013.

**Conference papers**

▷ **A. Fawzi***, S-M. Moosavi-Dezfooli*, P. Frossard, *Robustness of classifiers: from adversarial to random noise*, Neural Information Processing Systems (NIPS), 2016. (*: Equal contribution).
▷ **A. Fawzi**, P. Frossard, *Measuring the effect of nuisance variables on classifiers*, British Machine Vision Conference (BMVC), 2016. (*Oral presentation*)
▷ **A. Fawzi**, H. Samulowitz, D. Turaga, P. Frossard, *Adaptive data augmentation for image classification*, International Conference on Image Processing (ICIP), 2016.
▷ S-M. Moosavi-Dezfooli, **A. Fawzi**, P. Frossard, *DeepFool: a simple and accurate method to fool deep neural networks*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
▷ **A. Fawzi**, P. Frossard, *Manitest: Are classifiers really invariant?*, British Machine Vision Conference (BMVC), 2015. Matlab and C++ code available on project webpage.
▷ **A. Fawzi**, O. Fawzi, P. Frossard, *Fundamental limits on adversarial robustness*, ICML Deep Learning Workshop, 2015.
▷ **A. Fawzi**, P. Frossard, *Classification of unions of subspaces with sparse representations*, Asilomar Conference on Signals, Systems and Computers, 2013.
▷ **A. Fawzi**, P. Frossard, *A geometric framework for registration of sparse images*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2013.
▷ **A. Fawzi**, P. Frossard, *Thresholding-based reconstruction of compressed correlated signals*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2012.

## Conferences & workshops attended

- International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2012, Kyoto, Japan.
- International Traveling Workshop on Interactions between Sparse models and Technology (iTwist) 2012, Marseille, France.
- International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2013, Vancouver, Canada.
- Asilomar Conference on Signals, Systems and Computers, 2013, Monterey, CA, USA.
- ENS/INRIA Computer Vision and Machine Learning summer school (CVML) 2013, Paris, France.
- Signal Processing with Adaptive Sparse Structured Representation (SPARS) 2013, Lausanne, Switzerland.
- International Traveling Workshop on Interactions between Sparse models and Technology (iTwist) 2014, Namur, Belgium.
- International Conference on Machine Learning (ICML) 2015, Lille, France.
- International Computer Vision Summer School (ICVSS) 2015, Sicily, Italy.
- British Machine Vision Conference (BMVC) 2015, Swansea, UK.
- Neural Information Processing Systems (NIPS) 2015, Montreal, Canada.

## Work and research experience

**PhD thesis** *05/2012 - Ongoing*

The recent decade has witnessed the emergence of huge volumes of visual data. In this context, one of the key challenges is to classify data efficiently, accurately, while being resilient to corruptions. We propose in this thesis methods to diagnose and enhance classification schemes to be applied to real and possibly hostile environments.

**Internship IBM Thomas J Watson Research Center** *09/2015 - 12/2015*

▷ *Enhancing the robustness of classifiers using automatic data augmentation schemes*

State-of-the-art image classification systems are now reaching performances that are close to those of the human visual system in terms of accuracy on several datasets. However, such classifiers can be extremely unstable to small image corruptions. We seek to increase the robustness of such classifiers through automatic data augmentation procedures.

**Internship IBM Research Dublin** *02/2014 - 06/2014*

▷ *Multi-task additive models with shared transfer functions*

In this project, we develop a framework that extends additive regression models to multi-task scenarios, with multiple response variables. Assuming that *unknown correlations* exist between the different tasks, we propose a theory and algorithm that detects such correlations, and learn a *global* model for the tasks. We specifically focus on applications of the proposed framework for the forecasting of electric consumption data measured by smart meters.

▷ *Constrained additive index models*
In many large-scale forecasting problems, thousands of covariates, such as temperatures at different weather stations, are available. Fitting a model in this scenario is challenging as the model can hardly be *interpreted* by field experts, and tends to *overfit* the data. We introduce a constrained model that addresses these two issues, and we propose a principled fitting algorithm based on novel optimization techniques. We show applications of our method in electric load forecasting and bike prediction problems.

**Master's project in Signal Processing Laboratory, EPFL** *09/2011 - 02/2012*
▷ *Geometric group sparsity in image analysis*
This research project focuses on the approximation of sparse and structured signals in redundant dictionaries, where the structure is defined in a semantic way. Applications include image denoising and classification.

**Semester project in Signal Processing Laboratory, EPFL** *02/2011 - 06/2011*
▷ *Thresholding-based reconstruction of compressed correlated signals.*
Design of a low complexity joint decoder that exploits inter-sensor correlations to reconstruct signals from a few random measurements per sensor.

SERVICE
Reviewer for
- IEEE Transactions on Signal Processing
- IEEE Transactions on Image Processing
- IEEE Signal Processing Letters
- IEEE Pattern Recognition Letters
- Elsevier Digital Signal Processing (DSP)
- IEEE Transactions on Neural Networks and Learning Systems
- IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)

COMPUTER SKILLS
Matlab, C/C++, Java.

LANGUAGES
Bilingual French and Arabic
Fluent in English (TOEFL IBT score: 103, TOEIC score: 975)
Basic level in Spanish

CITIZENSHIP
French and Egyptian

DATE OF BIRTH
20/06/1989